Gustavo João Cunha Oliveira Ferreira

Ponto de Acesso Multifunções para Domótica

Gustavo João Cunha Oliveira Ferreira

Ponto de Acesso Multifunções para Domótica

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Professor Doutor Arnaldo Silva Rodrigues de Oliveira, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

O Júri / The Jury

Presidente / President	**Professor Doutor Alexandre Manuel Moutela Nunes da Mota** Professor Associado, Universidade de Aveiro
Arguente Principal / Main Examiner	**Professor Doutor José Paulo Oliveira Santos** Professor Auxiliar, Universidade de Aveiro
Orientador / Advisor	**Professor Doutor Arnaldo Silva Rodrigues de Oliveira** Professor Auxiliar, Universidade de Aveiro

Agradecimentos / Acknowledgements

Em primeiro lugar, pretendo agradecer ao Professor Doutor Arnaldo Oliveira, orientador desta dissertação, pela motivação que incutiu em mim no aprofundamento dos meus conhecimentos sobre a temática da domótica e pelo apoio dispensado na realização deste projeto.

Agradeço, também, aos meus pais e à minha irmã, a força e a coragem que sempre me transmitiram, especialmente nos momentos mais difíceis deste meu percurso académico.

Pretendo manifestar, ainda, o meu reconhecimento à Universidade de Aveiro, pelo acolhimento proporcionado e pela qualidade do ensino ministrado.

Palavras Chave

Domótica, Sistemas Embutidos, *Access Point* Integrado, *Ethernet*, *Raspberry Pi*.

Resumo

Os principais objetivos desta dissertação consistem em desenvolver um sistema que seja capaz de atingir um conjunto mais alargado de consumidores através da redução do preço e que seja bastante modular para permitir adições futuras de novos serviços a estruturas que já estejam implementadas e em funcionamento, sendo o interface com o utilizador baseado numa *web page* através da qual o utilizador comanda todos os serviços instalados na sua habitação.

No entanto, importa referir que, tendo em conta que as frequências de operação do *wireless* são cada vez mais elevadas, como é o caso do protocolo IEEE 802.11ad, que opera na gama dos 60GHz, será necessário utilizar mais *access points* por habitação, uma vez que as ondas eletromagnéticas são demasiado pequenas e, por isso, têm bastante dificuldade em penetrar paredes, pelo que, no limite, será necessário um *access point* por divisão.

A arquitetura de rede desenvolvida neste projeto é composta por um conjunto de programas da infraestrutura da rede (*discovery suite*, que inclui *discovery notifier* e *discovery center*, *intAPcom*, *intAPcomConv* e *web page*) que servem de suporte ao programa que opera o serviço de comando de estores (*digital I/O*: *Shutter Commander*). De salientar que, para provar a viabilidade do sistema, foi criado um protótipo através do qual se demonstra o correto funcionamento do serviço de comando de estores e, como tal, fica confirmada a eficácia do funcionamento da infraestrutura da rede tecnológica desenvolvida.

Assim, os objetivos propostos para a presente dissertação foram atingidos com sucesso, pois o facto de a solução desenvolvida ser baseada na comunicação via *Ethernet* (TCP/IP) permite aproveitar a cablagem existente ou comunicar através de *wireless*, reduzindo consideravelmente os custos de implementação, sendo também de destacar, por um lado, que a infraestrutura da rede tecnológica utilizada torna possível que todos os serviços funcionem de forma correta e independentemente da estrutura da habitação e/ou do número de serviços já implementados, o que aumenta a modularidade desta solução, e, por outro, que a interface utilizada é baseada numa *web page*, através da qual o utilizador comanda todos os serviços instalados na sua habitação, aumentando a comodidade proporcionada por esta solução.

Keywords Home Automation, Embedded Systems, Integrated Access Point, Ethernet, Raspberry Pi.

Abstract The main objectives of this dissertation are to develop a system that is able to reach a wider range of consumers by reducing the price and it is modular enough to allow for future additions of new services to structures that are already implemented and in operation, and the user interface based on a web page through which the user controls all the services installed in your home.

However, it should be noted that, the operating frequency of wireless technology are becoming higher, as is the case of the IEEE 802.11ad protocol, which operates in the range of 60GHz, and because of that more access points should be used in each house, since electromagnetic waves are too small and therefore have a hard time penetrating walls, ultimately, an access point per division is required.

The network architecture developed in this project consists of a set of programs of the network infrastructure (discovery suite, that includes discovery notifier and discovery center, intAPcom, intAPcomConv and web page) that support the program which operates the service that commands shutters (digital I/O: Shutter Commander). Stress that to prove the feasibility of a prototype system was set up whereby it demonstrates the correct operation of the Shutter Commander service and, as such, is proved correct operation of the developed technological infrastructure network.

Thus, the proposed objectives for this dissertation were achieved successfully, since the fact that the solution developed is based on communication via Ethernet (TCP/IP) allows leverage existing wiring or communicate via wireless, significantly reducing deployment costs, is also worth noting, first, that the infrastructure of the network technology used makes it possible that all services operate correctly and independently of the structure of housing and/or the number of services already in place, which increases the modularity of this solution, and secondly, that the interface used is based on a web page, through which the user controls all the services installed in your home, increasing the convenience provided by this solution.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

AJAX *Asynchronous JavaScript and XML*

AP *Access Point*

CPU *Central Processing Unit*

CSI *Camera Serial Interface*

DC *Direct Current*

DES *Data Encryption Standard*

DHCP *Dynamic Host Configuration Protocol*

DiffServ *Differentiated Services*

DNS *Domain Name System*

DSI *Display Serial Interface*

DSL *Digital Subscriber Line*

EHS *European Home Systems*

TIA/EIA *Telecommunications Industry Association / Electronic Industries Alliance*

EIB *European Installation Bus*

FDDI *Fiber Distributed Data Interface*

FTP *File Transfer Protocol*

GPU *Graphics Processing Unit*

GSM *Global System for Mobile*

HTML *HyperText Markup Language*

HTTP *Hypertext Transfer Protocol*

I2C *Inter-Integrated Circuit*

IAP *Integrated Access Point*

IBM *International Business Machines*

ID *Identifier*

IEC *International Electrotechnical Commission*

IEEE *Institute of Electrical and Electronics Engineers*

IETF *Internet Engineering Task Force*

IMAP *Internet Message Access Protocol*

I/O *Input/Output*

IP *Internet Protocol*

IPTV *Internet Protocol Television*

IPv4 *Internet Protocol version 4*

IPv6 *Internet Protocol version 6*

IR *Infrared*

IRDA *Infrared Data Association*

ISDN *Integrated Services Digital Network*

ISO *International Organization for Standardization*

JSON *JavaScript Object Notation*

LAN *Local Area Network*

LON *Local Operating Network*

MAC *Media Access Control*

MAN *Metropolitan Area Network*

MPLS *Multiprotocol Label Switching*

NETBIOS *Network Basic Input/Output System*

OSI *Open Systems Interconnection*

PAN *Personal Area Network*

PC *Personal Computer*

PDA *Personal Digital Assistant*

PL *Power Line*

PLC *Power Line Communication*

PoE *Power over Ethernet*

PoL *Power over Line*

POP *Post Office Protocol*

QoE *Quality of Experience*

QoS *Quality of Service*

RF *Radio Frequency*

RSVP *Resource Reservation Protocol*

SIP *Session Initiation Protocol*

SMTP *Simple Mail Transfer Protocol*

SONET *Synchronous Optical Networking*

SPI *Serial Peripheral Interface*

SSH *Secure Shell*

TCP *Transmission Control Protocol*

ToS *Type of Service*

UDP *User Datagram Protocol*

USB *Universal Serial Bus*

VLAN *Virtual LAN*

VOD *Video on Demand*

VoIP *Voice over IP*

WAN *Wide Area Network*

WLAN *Wireless LAN*

WMAN *Wireless MAN*

WPAN *Wireless PAN*

Capítulo 1

Introdução

Neste capítulo apresentam-se conceitos de domótica e suas principais funcionalidades, bem como a motivação, objetivos e estrutura da presente dissertação.

1.1 Enquadramento

O presente trabalho enquadra-se no contexto da domótica, a qual é uma tecnologia que permite a gestão integrada dos recursos habitacionais. O termo domótica resulta da junção de *domus* (palavra latina que significa casa) com robótica (controlo automatizado de eventos sem um pensamento consciente).

De referir que as funcionalidades mais importantes num sistema domótico são [1]:

- a automatização doméstica com vista a um maior conforto;
- a segurança de modo a detetar intrusões;
- a proteção com o intuito de detetar incêndios e fugas de gás/água, entre outros; e,
- a monitorização dos consumos e gestão de energia.

Ao conceito de domótica estão associadas as designações *Home Automation*, *Smart Home* e *Intelligent Home*. De notar que existe um paralelo entre domótica e edifícios inteligentes, que apresentam objetivos comuns embora com ênfases diferentes, uma vez que num edifício o aspeto mais relevante é a gestão de recursos energéticos, enquanto numa habitação o aspeto mais importante é o conforto [1]. Enquanto os edifícios são funcionalmente mais complexos e envolvem um elevado número de utilizadores, requerendo que a gestão esteja a cargo de pessoas especializadas, as habitações possuem um reduzido número de utilizadores, que normalmente são responsáveis por gerir o sistema, sendo, por isso, expectável que os sistemas domóticos sejam o mais simples possível [1].

No final dos anos 70, a aplicação de conceitos semelhantes à domótica estava direcionada para contextos militares, complexos industriais e grandes edifícios de escritórios. Porém, com o passar dos anos tem-se assistido ao desenvolvimento de soluções cada vez mais orientadas para o contexto doméstico. A domótica apresenta um futuro promissor no mercado imobiliário, sendo de destacar, por um lado, o ponto de vista do cliente final, que procura soluções para os seus problemas e necessidades em habitação, e, por outro, o ponto de vista das empresas imobiliárias, que recorrem aos serviços da domótica

para agregarem valor ao imóvel através da implementação de soluções que atendam às expectativas do cliente [2].

1.2 Motivação

Nos próximos anos vai seguramente assistir-se a uma procura crescente de soluções inovadoras, versáteis, económicas, de alto desempenho e fácil utilização para aplicações domóticas. As motivações para dotar as habitações com algum grau de inteligência resultam dos requisitos crescentes ao nível do conforto, segurança, entretenimento, eficiência e autonomia energética, utilização racional de recursos hídricos, acesso ubíquo a diferentes tipos de serviços de informação com necessidades de qualidade de serviço e/ou de experiência. Apesar da existência de um número considerável de normas, quer de domínio público, quer proprietárias, a oferta de produtos nesta área é relativamente reduzida e os disponíveis atualmente apresentam ainda um elevado custo de instalação, manutenção e atualização, restringindo, ao contrário do que seria desejável, estas soluções a nichos e segmentos altos do mercado. No entanto, a quantidade e diversidade de módulos computacionais, sensoriais e de comuniação disponíveis no mercado, a preços cada vez mais acessíveis, tem facilitado o desenvolvimento de soluções tecnológicas neste domínio.

O trabalho desenvolvido nesta dissertação pretende reduzir o custo da implementação destes serviços e permitir que qualquer habitação, mesmo aquelas que já estão construídas, possam beneficiar do conforto e da comodidade que a domótica pode proporcionar aos seus utilizadores.

Tendo em conta que nos últimos anos se tem assistido, por um lado, a um crescimento do interesse pela domótica por parte da população em geral e, por outro, a uma aposta no desenvolvimento da domótica principalmente por parte das empresas imobiliárias, surge a oportunidade de criar um sistema que permita implementar numa habitação, um conjunto de serviços de forma simples, eficiente e económica.

1.3 Objetivos

Os principais objetivos do presente trabalho de dissertação de mestrado consistem no desenvolvimento de um sistema:

- que permita satisfazer as necessidades de um conjunto mais alargado de consumidores e que, por isso, possa beneficiar de uma penetração no mercado superior à dos produtos já existentes na área da domótica. Neste sentido, ao reduzir os custos de implementação dos serviços, é alargada a oferta a classes sociais mais baixas, com a possibilidade de implementar estes serviços em habitações já construídas, abrindo caminho para que qualquer consumidor possa instalar um novo serviço na sua habitação.

- bastante modular, que permita adições futuras de novos serviços a estruturas que já estejam implementadas e em funcionamento. Para tal, foi criada uma *suite* de protocolos

e *software*, na qual todos os serviços assentem, para que possam funcionar corretamente e independentemente da estrutura da habitação e/ou do número de serviços já implementados.

- em que a interface com o utilizador seja baseada numa *web page*, através da qual o utilizar comanda todos os serviços instalados na sua habitação. A *web page* pode ser acedida por qualquer dispositivo que contenha um *web browser*, desde que esse dispositivo esteja ligado à rede dessa habitação. Todas as comunicações são feitas através do protocolo *Ethernet*, podendo comunicar por cabo, se a habitação tiver a cablagem já instalada, ou por *wireless*, se não houver qualquer cablagem instalada.

A abordagem adotada nesta dissertação baseia-se na extensão de um *access point* de forma a integrar os dispositivos e os mecanismos de suporte à domótica e a sua conexão à rede de dados cablada e universal. Neste sentido, as funções do *Integrated Access Point* para sistemas domóticos são permitir a montante a utilização de uma rede de dados cablada, disponibilizar a jusante uma rede de dados não cablada e integrar os componentes e mecanismos para oferecer serviços adicionais e respetivas interfaces para acesso à rede. A escolha de um equipamento do tipo *access point* para integrar funcionalidades de aplicações domóticas é justificada pelo facto de se acreditar que, num futuro próximo, para garantir uma boa cobertura e uma elevada largura de banda, será necessário instalar múltiplos *access points* numa habitação. Por fim, importa referir que a modularidade do *Integrated Access Point* é um aspeto fundamental de modo a permitir a seleção "à la carte"das funcionalidades pretendidas.

1.4 Estrutura da Dissertação

Esta secção pretende apresentar a estrutura da presente dissertação, fazendo um breve resumo dos conteúdos abordados em cada um dos capítulos.

Capítulo 1 - Introdução - enquadra o tema da domótica no contexto da literatura existente, fazendo ainda referência à motivação que levou ao desenvolvimento deste trabalho e aos objetivos que se pretendem alcançar.

Capítulo 2 - Revisão de Literatura - descreve o estado da arte do tema da domótica, destacando algumas aplicações da domótica, e apresenta os principais conceitos de redes, evidenciando a importância da convergência tecnológica e da computação em nuvem.

Capítulo 3 - Especificação do Sistema - mostra a arquitetura do sistema implementado e os serviços que interagem com o mesmo, apresentando a sua modularidade e a forma de interação entre o utilizador e o sistema, que se baseia numa interface *web*.

Capítulo 4 - Implementação do Sistema - explica as metodologias e ferramentas usadas para o desenvolvimento da infraestrutura criada no âmbito desta dissertação, bem como o raciocínio utilizado na implementação do *Integrated Access Point*

(IAP), descrevendo o funcionamento, quer dos programas desenvolvidos para a Infraestrutura da Rede Tecnológica (IRT), quer do programa de suporte à operacionalização do serviço: *Shutter commander* (Sc).

Capítulo 5 - Demonstração do Sistema - evidencia os resultados do trabalho desenvolvido, através da apresentação de um caso de estudo, bem como a configuração do sistema *Linux*, que permite que este funcione como um *access point*, e a configuração do servidor *OpenVPN*.

Capítulo 6 - Conclusões e Trabalho Futuro - expõe as principais conclusões, destacando a contribuição científica do trabalho desenvolvido, bem como alguns tópicos de possíveis desenvolvimentos futuros.

Capítulo 2

Revisão de Literatura

Neste capítulo apresenta-se, por um lado, o estado da arte do tema da domótica, com destaque para algumas aplicações já existentes, e, por outro, os principais conceitos de redes, evidenciando a convergência tecnológica e a computação em nuvem.

2.1 Conceitos de Domótica

O surgimento da domótica, em meados da década de 70 do século XX, está relacionado com a necessidade de controlar a iluminação, a climatização e a segurança, entre outros, visando a gestão integrada de todos os recursos das instalações, nomeadamente em contexto militar, industrial e empresarial.

Atualmente, embora com a mesma ideia subjacente, foi alterado o contexto para o qual o sistema está pensado, tendo sido estendida a sua aplicação na procura de soluções orientadas para contextos domésticos e ambiente urbano como, por exemplo, iluminação pública nas cidades.

Com a domótica aplicada a contextos domésticos procura-se usar dispositivos para automatizar as rotinas e tarefas diárias de uma habitação, sendo ainda necessário reduzir o custo de instalação, manutenção e atualização, o que requer a diminuição do número de redes e da quantidade de cablagem utilizada para assegurar os diferentes tipos de serviços.

As principais rotinas e tarefas a automatizar, dentro de uma solução completa, são:

- controlo de temperatura ambiente (aquecimento, ventilação e ar condicionado);
- iluminação e, difusão de áudio e vídeo;
- controlo de portas e estores;
- alarmes, vigilância e deteção de intrusão;
- videoporteiro e intercomunicador;
- controlo de rega automática;
- telefone e VoIP; e,
- autossuficiência energética.

Tendo em conta que estas soluções são centralizadas, apenas é necessário um comando para controlar todos os sistemas.

2.1.1 Origem e Evolução da Domótica

O aparecimento da eletricidade permitiu elevar o nível de conforto nas nossas habitações e, consequentemente, dar início à utilização dos eletrodomésticos (décadas de 50/60 do século XX), tais como máquinas de lavar louça e roupa, frigorífico, forno elétrico, placas vitrocerâmicas, batedeira elétrica, micro-ondas, entre outros.

Estes equipamentos, que permitiram aumentar os níveis de conforto das famílias, não existiriam sem o desenvolvimento da eletrónica, que permite realizar programações (rotinas), que regulam cada processo como, por exemplo, lavar a roupa ou gravar um vídeo.

Posteriormente surgiu a domótica, que se ocupa da integração e regulação de ambos os sistemas (elétrico e eletrónico), de tal maneira que a habitação é capaz de "sentir" (detetar a presença de pessoas, medir a temperatura ou a intensidade da luz) e "reagir" por si só a estes estímulos (regulando a temperatura e a iluminação ou ligando o alarme), ao mesmo tempo que é capaz de "comunicar" e "interagir" connosco (telecontrolo) utilizando múltiplos meios (computador, telemóvel ou *tablet*), atingindo elevados níveis de conforto, segurança e, sobretudo, poupança energética, conforme ilustrado na figura 2.1.

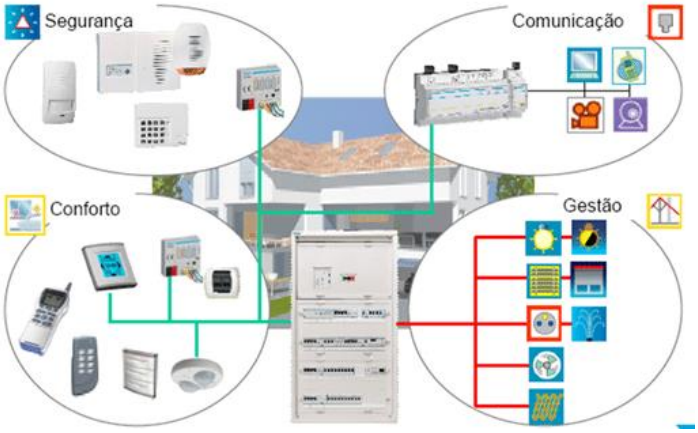


Figura 2.1: Telecontrolo de uma Instalação (extraído de [3]).

A evolução da domótica compreende uma série de etapas, desde os primeiros protocolos orientados ao controlo remoto, até aos protocolos mais complexos capazes de realizar funções lógicas complexas.

Surgimento em Finais dos Anos 70 do Século XX: o Protocolo X10

A história da domótica foi iniciada com o protocolo X10, em 1975, criado para o telecontrolo e baseado em *power line*, conforme apresentado na figura 2.2, sendo os sinais constituídos por pequenos impulsos que são codificados numa onda portadora de

120 KHz e que é transmitida quando a corrente alternada (de 60 Hz) passa por zero [4]. Um bit é transmitido por cada passagem por zero. Este protocolo desenvolveu-se muito nos Estados Unidos e na Europa, sobretudo no Reino Unido e em Espanha.

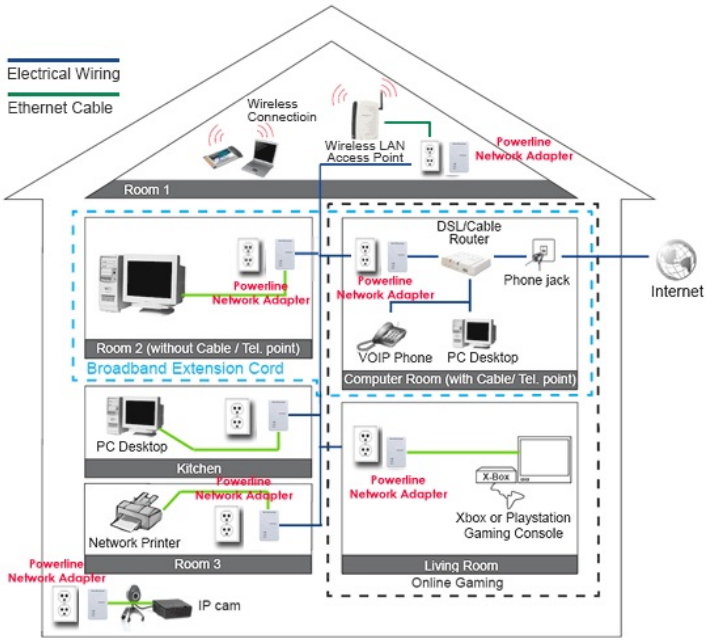


Figura 2.2: *Power Line* (extraído de [6]).

O protocolo X10 é um dos protocolos mais antigos e baratos no mercado da domótica, pelo que existem muitos dispositivos no mercado. Esta tecnologia ainda continua no mercado apesar da grande concorrência que enfrenta de novos *standard*. Uma das vantagens do protocolo X10 traduz-se na possibilidade de usar as linhas de energia elétrica, como meio de comunicação [7]. No entanto, a transmissão de mensagens é feita sequencialmente com um comando, sendo enviado um de cada vez, o que constitui uma das grandes desvantagens do X10, uma vez que vários dispositivos podem enviar sinais concorrentemente, originando problemas de controlo no acesso ao meio [7].

Ao utilizar a corrente elétrica para transmitir o sinal, o protocolo X10 fica dependente do potencial ruído envolvente, o que influencia diretamente a qualidade do sinal, pois, embora existam filtros que permitem atenuar o ruído inserido no sinal, nunca conseguem irradicá-lo totalmente. Para além disso, a atenuação do sinal limita a distância máxima entre dois equipamentos [4]. O protocolo X10 apresenta uma fiabilidade reduzida e, apenas permite controlar sistemas com regulações simples (*on/off*) ou de *dimmer*, ignorando os problemas relacionados com funções lógicas mais complexas como, por exemplo, a climatização.

Nascimento dos Protocolos *Standard*: KNX e LON

Os protocolos mais utilizados na domótica são os protocolos *standard* KNX e LON.

KNX

A história de sucesso do Konnex (KNX) começou em maio de 1990, com a criação da *European Installation Bus Association* (EIBA), constituída por 15 construtores europeus da indústria da domótica. O objetivo da EIBA era distribuir e promover o chamado *European Installation Bus* como um sistema *standard* internacional. Em maio de 1999, a EIBA associou-se à *BatiBUS Club International* e à *European Home Systems Association* (EHSA) formando a *Konnex Association* [8].

O KNX foi standardizado pela CENELEC em dezembro de 2003. No ano de 2006, parte significativa deste *standard* (EN 50090) foi incluído no *standard* internacional ISO/IEC 14543, fazendo do KNX o único *standard* aberto a nível mundial para domótica. Neste contexto, aberto significa que dispositivos de fabricantes diferentes podem comunicar uns com os outros através do KNX. O *standard EN 50090* detalha as características e configurações do sistema KNX e define as regras de topologia para as ligações do barramento, bem como os protocolos que especificam a forma como os dispositivos comunicam uns com os outros [8].

A figura 2.3 apresenta exemplos de aplicações do protocolo KNX.

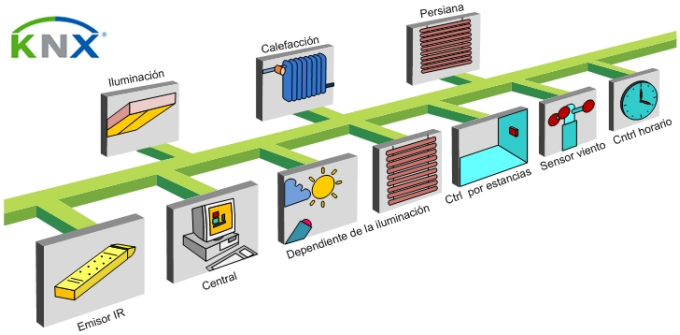


Figura 2.3: Protocolo KNX (extraído de [9]).

O KNX foi desenvolvido para ser usado nos principais sistemas de domótica e permite o planeamento, a implementação e a integração de sistemas individuais como uma rede interligada. Todos os dispositivos dos vários sistemas têm interfaces *standard* do tipo KNX, para que possam comunicar uns com os outros, simplificando o planeamento e a implementação dos sistemas domóticos e proporcionando mais funcionalidades, flexibilidade e conforto. Uma vez que cada dispositivo tem o seu próprio microcontrolador, não é necessária a existência de uma central de controlo [8].

Para que os dispositivos executem as suas funções, basta definir os parâmetros corretos, que podem ser alterados a qualquer altura, tornando o KNX extremamente flexível e permitindo que o sistema se ajuste e expanda para atender a novos requisitos. O KNX pode ser usado para controlar automaticamente o sistema de aquecimento, as luzes, o ar condicionado e os sistemas de segurança, entre outros [8].

Os produtos KNX têm custos mais elevados do que os usados em instalações convencionais. Geralmente, o investimento só se justifica se for para conectar vários sistemas ou se uma instalação precisar de ser suficientemente flexível para que possa ser rápida

e eficazmente modificada, de forma a atender a futuros requisitos [8].

LON

O *LonWorks* é uma solução aberta para domótica e controlo de redes, tendo sido desenvolvido pela empresa americana *Echelon* e desenhado de forma a ser usado com controladores domóticos centralizados ou com componentes de domótica descentralizados [8].

O componente principal do *Local Operating Network* (LON) é um microcontrolador designado por *Neuron Chip*, que foi introduzido no mercado em 1990. O *LonWorks* é um sistema de barramento *standard* que permite a comunicação entre dispositivos inteligentes, através de uma rede local, estando as suas especificações publicadas no ANSI/CEA-709.1, no ANSI/EIA-852 e no ISO/IEC DIS 14908 [8].

A figura 2.4 apresenta exemplos de aplicações do protocolo LON.

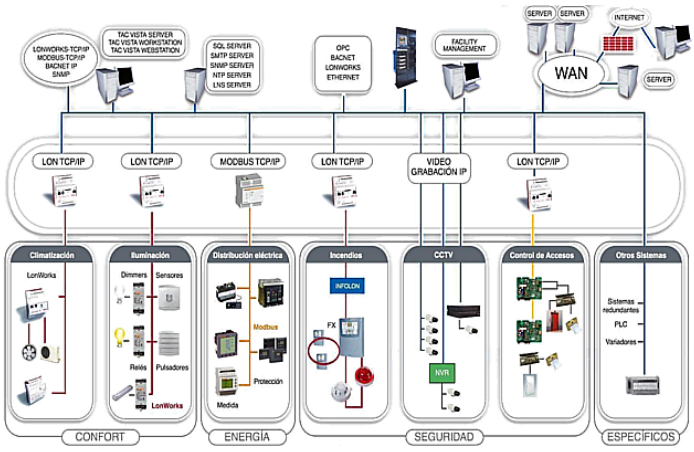


Figura 2.4: Protocolo LON (extraído de [10]).

A tecnologia LON tornou-se parte integrante no controlo de habitações, focando-se principalmente em automação de divisões. Esta tecnologia é usada ainda na indústria, em edifícios comerciais e na rede de controlo de transportes a nível internacional [8].

Expansão dos Protocolos *Standard* e Aparecimento de Sistemas sem Fios

A partir de 2006, começaram a surgir protocolos que competem diretamente com o X10, nomeadamente o *Insteon*, o *Z-wave*, o *Zigbee* e o *EnOcean* [7].

Insteon

O *Insteon* foi desenvolvido em 2004 pela *SmartLabs* [4]. Esta tecnologia foi desenhada para integrar sistemas *power line* com sistemas *wireless*, tendo sido desenvolvida para substituir o protocolo *standard* X10, de forma a permitir que os dispositivos, sensores ou interrutores, possam ser utilizados de forma conjunta em sistemas *power line* e/ou rádio frequência. Para além do protocolo X10, esta é a única tecnologia que comunica tanto por *wireless* como por *power line* [7]. É de salientar, ainda, o facto de ser uma tecnologia parcialmente compatível com dispositivos X10. Os comandos do X10 e

do *Insteon* não são similares, mas o *chipset* do *Insteon* tem a capacidade de responder a mensagens do X10 e, por isso, consegue comunicar com os seus dispositivos [4].

Z-Wave

O *Z-Wave* foi desenvolvido em 2004 por uma empresa dinamarquesa, denominada Zensys, e utiliza um protocolo de comunicação baseado em RF para sinalização e controlo. Assente no conceito do *ZigBee*, o *Z-Wave* tinha como objetivo construir uma tecnologia mais simples e barata [7].

O *Z-Wave* é a tecnologia mais usada em sistemas domóticos e a mais aceite no mercado, oferecendo fiabilidade e estabilidade. A caraterística mais importante dos dispositivos Z-Wave é a sua compatibilidade com os diferentes sistemas das outras marcas. O Z-Wave é um protocolo de rede de malha e, por isso, os dispositivos podem comunicar entre si. Cada dispositivo Z-Wave tem um ID de rede único e cada rede tem um único ID, criando um sistema seguro [4].

Zigbee

O *ZigBee* é um *standard* IEEE 802.15.4 usado na domótica, que foi aceite e ratificado pela *ZigBee Alliance* em dezembro de 2004. O *ZigBee* opera como uma rede de sinalização que se assemelha à tecnologia *Bluetooth* e *WiFi* [7].

Os dispositivos *ZigBee* são atrativos porque são de baixo consumo e com especificações abertas, o que os torna ideais para aparelhos que usam baterias. O *ZigBee*, tal como o *Z-Wave*, é um protocolo de rede de malha, onde os dispositivos podem comunicar uns com os outros e podem agir como *repeaters* [4].

EnOcean

O *EnOcean* é uma das tecnologias mais recentes na domótica, essencialmente focada na chamada *zero energy comsumption*, através do uso de *energy harvesting*. Esta tecnologia tem a caraterística única de funcionar sem baterias e, mesmo assim, dispõe da capacidade de comunicar por *wireless*. Esta capacidade é conseguida através de conversores de micro energia em conjunto com eletrónica de muito baixa potência [4].

Futuro dos Sistemas Domóticos

A tendência é que os protocolos *standard* prossigam ligados ao protocolo IP, sendo este o meio de transporte de ambos, *KNX* e *LON*. No contexto da domótica merecem destaque os seguintes protocolos *standard* [5]:

- O *KNXnet/IP* é usado, principalmente, para controlo da habitação, sendo bastante conhecido, apresenta uma estrutura de dados bastante simples e possui vários produtos no mercado. Este protocolo tem ferramentas comuns para a sua instalação e configuração, acessíveis a quase todos os instaladores. Contudo, atualmente esta tecnologia é apenas usada no *backbone* de comunicações ou para propósitos de configuração.
- O *BACnet IP* é usado, principalmente, em aplicações HVAC, contém um conjunto de perfis bem definidos para as aplicações e possui inúmeros produtos no mercado.

No entanto, apresenta uma estrutura de dados complexa, não disponibilizando ferramentas comuns para a sua instalação e configuração, pelo que a instalação só pode ser feita por instaladores experientes.

- O *LON over IP* é usado para sistemas HVAC e controlo da habitação, contém um conjunto de perfis definidos para aplicações, apresenta uma estrutura de dados mais simples do que o *BACnet IP*, disponibiliza ferramentas para a sua instalação e configuração, pelo que a instalação é mais simples do que o *BACnet* e mais complexa do que o *KNX*. Porém, atualmente existem poucos produtos disponíveis no mercado.

2.1.2 Mecanismos de Suporte à Domótica e Conexão à Rede de Dados

A domótica utiliza vários elementos de uma forma sistemática, aliando as vantagens dos meios eletrónicos aos informáticos, de forma a obter uma utilização e gestão integrada dos diversos equipamentos de uma habitação. Neste sentido, a domótica vem tornar a vida mais confortável e mais segura, na medida em que permite que as tarefas mais rotineiras e aborrecidas sejam executadas automaticamente.

De notar que a utilização pode ser mais ou menos automática, sendo que, nos sistemas passivos, o dispositivo reage só quando lhe é transmitida uma ordem, dada diretamente pelo utilizador através de um interrutor ou um comando, que pode ser uma ordem ou um conjunto de ordens macro. Em sistemas mais avançados, o dispositivo não só interpreta parâmetros introduzidos pelo utilizador, como reage às circunstâncias percecionadas através de sensores, tais como detetar que uma janela está aberta e avisar o utilizador, registar que a temperatura está a diminuir e ligar o aquecimento, identificar a velocidade do vento e enviar uma ordem para fechar as janelas, entre outras.

Deste modo, a domótica veio permitir o controlo e monitorização, à distância, das principais funcionalidades da habitação, sendo apenas necessário acesso à *internet*.

Algumas Aplicações da Domótica

De seguida são apresentados alguns exemplos dos diferentes tipos de aplicações da domótica:

- Automação: consiste em programar tarefas diárias, individuais ou em complexos conjuntos, para serem realizadas de uma forma automática, sem a intervenção do homem, permitindo reduzir o tempo gasto em rotinas e evitar esquecimentos.

- Iluminação: consiste na gestão dos gastos de eletricidade através das funções de regulação de intensidade, otimizando o consumo de energia, tendo em conta a presença/ausência, hábitos e horários.

- Cinema em casa: consiste na junção de todos os comandos do sistema de *home cinema* e na criação de macros para realizar as tarefas simples como sejam as de preparar o ambiente para assistir a um filme pré-programado de acordo com os utilizadores da habitação.

- Som ambiente: consiste na instalação de colunas em determinadas divisões ligadas a uma central de som, sendo possível cada utilizador escolher o que pretende ouvir na divisão em que está presente, podendo ainda partilhar a música do seu dispositivo portátil, telemóvel ou leitor de MP3, para a rede doméstica.
- Climatização: consiste na programação de horários para ativar/desativar equipamentos de aquecimento, ventilação ou ar condicionado, permitindo manter um bom nível de conforto e poupando energia, para além de proporcionar a comodidade de, a partir do exterior da habitação, poder certificar-se de que realmente o aquecimento se encontra desligado.
- Segurança: consiste na utilização de sensores para detetar possíveis intrusos, denunciando a sua presença, detetar fugas de gás, inundações ou incêndios em fase inicial, avisando do sucedido de forma a serem tomadas providências, quer pelos proprietários, quer pelos profissionais de manutenção ou bombeiros.

Internet das Coisas

A *internet* tem tido uma forte evolução desde que foi criada, começando como uma rede de computadores para uso académico, usada apenas por alguns, hoje, é universal e uma fonte de informação utilizada por todos. No entanto, a comunicação através da *internet* já não é predominantemente iniciada pelo ser humano à procura de informação. Cada vez mais, os denominados *smart objects* como, por exemplo, os *smartphones*, iniciam comunicações independentemente do utilizador para obter informação sobre os mais variados campos [11].

A *Internet of Things* (IoT), também conhecida por *Internet of Objects*, pode ser considerada como a nova revolução tecnológica. No entanto, existem várias barreiras que ameaçam abrandar o desenvolvimento da IoT, entre as quais, a transição para IPv6, a necessidade de ter um conjunto de *standard* e de desenvolver fontes de alimentação para milhões, ou até mesmo milhares de milhões de pequenos sensores [11].

O conceito da IoT surgiu no MIT, através de um grupo de trabalho dos laboratórios *Auto-ID*, originalmente denominado de *Auto-ID center*, que foi fundado em 1999. Este grupo trabalhava no campo de redes RFID e de tecnologias de sensores emergentes [11].

A visão da IoT é a de fundir o meio físico e o mundo digital, juntando os diferentes conceitos e componentes técnicos. A IoT visa criar uma rede de milhares de milhões de dispositivos, ligados por *wireless*, que comuniquem entre si e que sejam passíveis de serem identificados através de UID's (*Unique ID*). Esta visão promete criar um novo ecossistema em que os dispositivos inteligentes sejam capazes de se adaptarem ao seu ambiente, auto configurarem, auto manterem, auto repararem e, eventualmente, participarem ativamente na sua destruição, quando chegarem ao seu fim de vida. Estes dispositivos/objetos seriam capazes de coletar a energia necessária para se sustentarem, adaptarem às mudanças de ambiente e, ainda, lidarem com circunstâncias imprevistas. A IoT promete melhorar a qualidade de vida do ser humano [12].

Atualmente, a IoT é composta por um conjunto disperso de dispositivos com fins muito específicos. Os carros de hoje em dia, por exemplo, contêm múltiplos microprocessadores para controlar as funções do motor, as componentes de segurança e os sistemas de

comunicação, entre outros. Os edifícios comerciais e residenciais também têm vários sistemas para aquecimento, ventilação, ar condicionado, serviços de telefone, segurança e controlo de luz. Com a evolução da IoT, estas redes e muitas outras serão interligadas com segurança adicional, com capacidades de análise e gestão melhoradas [11].

Quando ultrapassámos a marca de ter mais objetos ligados à *internet* do que pessoas, uma enorme janela de oportunidades abriu-se para a criação de aplicações nas áreas de automação, sensorial e comunicação máquina a máquina, também conhecida por M2M (*Machine-to-Machine*) [11]. A inovação que a IoT representa, traz uma nova dimensão aos modelos de negócio já existentes em todos os setores. Por exemplo, as designadas *smart cities* têm como objetivo tornar a infraestrutura do serviço público e o processo de negócio mais inteligente, mais eficiente e mais sustentável, através da integração mais próxima da *internet* e da capacidade computacional. Sensores espalhados pela cidade, recolhem informação variada sobre a qualidade do ar, bens consumíveis, instalações usadas e outras informações relevantes para a vivência da comunidade. Esta informação pode depois ser analisada para tomar as medidas apropriadas para melhorar a qualidade de vida da cidade [12].

Em conclusão, a IoT representa uma importante evolução da *internet*. Se o ser humano for capaz de transformar os dados em informação, conhecimento e sabedoria, a IoT tem o potencial de modificar o mundo [11].

2.2 Conceitos de Redes

Nesta secção destaca-se a arquitetura de redes e o *Quality of Service*, bem como as tecnologias que permitem transmitir energia elétrica e dados pelo mesmo cabo, nomeadamente o *Power over Ethernet* e o *Power Line Communication*.

Por fim, são também abordados os temas da convergência tecnológica, que designa uma única infraestrutura para fornecer serviços que, anteriormente, requeriam equipamentos, canais de comunicação, protocolos e padrões independentes, e da computação em nuvem, que permite a partir de qualquer computador e em qualquer lugar do mundo aceder a informações, arquivos e programas num sistema único e independente da plataforma utilizada.

2.2.1 Arquitetura de Rede

O modelo designado por arquitetura de rede é composto por um conjunto de *standard*, sendo que cada documento descreve uma função específica da rede e, em conjunto, estes documentos definem todos os comportamentos que devem existir para o correto funcionamento da rede [13]. Alguns documentos definem protocolos, que são um conjunto de regras lógicas que os dispositivos devem seguir para conseguirem comunicar entre si, e outros documentos definem os requisitos físicos para a rede.

Os principais modelos de rede são [14]:

- Arquitetura ponto-a-ponto, em que cada utilizador gere o seu próprio computador, determinando o que quer partilhar, e todos os computadores pessoais são tanto servidor como cliente. Não existem servidores dedicados, nem administrador de redes,

pelo que são redes de baixo custo e de pequena dimensão, em que a segurança não é um fator importante; e,

- Arquitetura cliente-servidor, em que o cliente é uma estação de trabalho e o servidor é um computador dedicado, que possui informações centralizadas cuja única função é dar resposta aos pedidos dos clientes. Existe um servidor dedicado e um administrador de redes, a dimensão da rede é grande e a segurança é de primordial importância.

A grande desvantagem que as redes ponto-a-ponto apresentam em relação às redes cliente-servidor é a sua dificuldade em gerir os serviços, já que não existe um sistema operacional que centralize a administração da rede. Também não é possível estendê-las excessivamente, já que um número elevado de nodos sobrecarregaria o fluxo de dados tornando-a lenta e, consequentemente, ineficaz. A distribuição de funções cliente-servidor oferece vantagens substanciais, mas também apresenta custos elevados, pois apesar da agregação de recursos em servidores trazer maior segurança, acesso mais simples e controlo coordenado, o servidor introduz na rede um ponto de falha único, isto é, a rede não funciona sem um servidor operacional [14].

Importa, ainda, distinguir os seguintes conceitos: *internet*, *intranet* e *extranet*.

A *internet* é uma rede mundial de computadores, que interliga várias redes em todo o mundo utilizando os mesmos padrões de comunicação.

A *intranet* é um tipo especial de *internet*, cujo acesso é restrito a utilizadores de uma rede corporativa ou habitação, combinando o melhor da tecnologia cliente-servidor com o melhor da *internet*. Aumenta a produtividade através da partilha de documentos comuns, reduzindo distâncias através de reuniões virtuais e permitindo o acesso rápido a relatórios pelos tomadores de decisões.

A *extranet* é a parte da *intranet* que fica disponível na *internet* para acesso ao público em geral, com acesso controlado a algumas áreas.

A rede de computadores usa um modelo de rede designado *Transmission Control Protocol/Internet Protocol* (TCP/IP). O modelo TCP/IP define e referencia um conjunto de protocolos que permitem que os computadores comuniquem entre si.

Modelo TCP/IP

Investigadores de várias universidades contribuiram, voluntariamente, para os protocolos que foram desenvolvidos em torno de um trabalho realizado pelo departamento de defesa dos Estados Unidos, relacionado com a criação de um modelo de rede público, aberto e *vendor-neutral*. Estes esforços resultaram num modelo de rede competitivo, designado TCP/IP. Atualmente, o modelo TCP/IP é o mais usado, tendo substituído a maioria dos modelos proprietários. De referir que o modelo OSI, cujo desenvolvimento sofreu dada a lentidão do processo de padronização, quando comparado com o TCP/IP, nunca teve sucesso no mercado [13].

O TCP/IP é um padrão de comunicação entre diferentes equipamentos de comunicação de dados e diferentes sistemas operacionais e aplicativos, sendo que a sua arquitetura se traduz num conjunto de padrões e protocolos de comunicação de dados, utilizado

na interconexão e endereçamento de computadores e redes. O TCP/IP resulta da combinação de dois protocolos [15]:

- TCP (*Transmission Control Protocol*), que é responsável pelo controlo e qualidade da comunicação entre a origem (transmissor) e o destino (recetor); e,
- IP (*Internet Protocol*), que é resposável pelo endereçamento nas redes de modo a que os dados cheguem ao seu destino de acordo com o endereço de rede fornecido.

A arquitetura TCP/IP é uma arquitetura cliente-servidor, que se tornou padrão na comunicação entre redes e sistemas de informação em redes, quer para a conexão de computadores em redes locais, quer para conexão de redes remotas e distantes entre si. A *internet* pode ser definida como um sistema global de comunicações, que se encontra lógicamente interconectado pelo endereçamento, globalmente unívoco, do protocolo IP, sendo capaz de suportar comunicações com o auxílio da pilha de protocolos TCP/IP e tornar acessíveis serviços de comunicações de forma pública ou privada [15].

Nos apêndices A e B são apresentados os conceitos de topologias de rede e de redes de computadores, respetivamente.

2.2.2 Qualidade de Serviço

O *Quality of Service* (QoS) é a capacidade de garantir um certo nível de desempenho para um fluxo de dados ou de gerir os pacotes num ambiente de congestionamento, atribuindo diferentes prioridades a diferentes aplicações, utilizadores ou fluxos de dados [16].

A *internet* prevê protocolos de prevenção de congestionamento como, por exemplo, o *Transmission Control Protocol* (TCP), para reduzir o tráfego em condições de congestionamento. Aplicações de *Quality of Service*, tais como VoIP e IPTV, por requererem uma largura de banda constante e em grande parte de baixa latência, não podem usar TCP e não podem de outra forma reduzir a sua taxa de tráfego para ajudar a evitar o congestionamento. O *Quality of Service* permite criar filas com atributos de prioridade, para que se possa enviar primeiro o tráfego que requer baixa latência. Assim, o *Quality of Service* torna-se uma parte indispensável da capacidade da rede para lidar com um *mix* de tráfego, quer seja sensível ao tempo, quer possa ser enviado em *best effort*.

Para medir, quantitativamente, a qualidade do serviço são considerados vários aspetos relacionados com o serviço da rede, tais como as taxas de erro, a largura de banda, o atraso, a disponibilidade, o *jitter* (flutuação ou variação do sinal), a perda de pacotes e a entrega desordenada, entre outros. Por exemplo, para que as comunicações de voz e vídeo funcionem corretamente, a largura de banda deve ser o mais alta possível e o atraso, o *jitter* e a perda de pacotes devem ser os mais baixos possível [16].

A maioria das redes existentes foram concebidas para aplicações de dados que não necessitam de transmitir dados em tempo real. No entanto, como os dados de áudio e vídeo têm de ser recebidos em tempo real, é necessário considerar o *Quality of Service* no desenho da rede. Assim, na arquitetura de uma rede IP, para áudio e vídeo, devem ser considerados três conceitos fundamentais que afetam a transmissão de dados em tempo real, designadamente *network provisioning*, *queuing* e *classifying* [16].

Atualmente, a abordagem mais comum para obter *Quality of Service* é o *over provisioning*, que consiste em alocar mais largura de banda ou capacidade do que o necessário para o funcionamento de todas as aplicações de áudio, vídeo e dados regulares que são executados através da rede. Em regra, a largura de banda máxima exigida para o conjunto de todas as aplicações, incluindo áudio e vídeo, não deve exceder 75% da largura de banda da rede disponível. Como tal, em certa medida, é necessário *over provisioning* da rede, muito embora, por si só, não seja suficiente para garantir uma adequada qualidade de serviço [16].

Os engenheiros de redes têm vindo a constatar que a questão chave do *Quality of Service* é o *buffering* e não a largura de banda. Os *buffers* de transmissão nos *switches* e *routers* tendem a encher rapidamente, em redes de alta largura de banda, causando o descarte de pacotes, que por sua vez causam *clipping* e *skipping* no áudio e vídeo. No entanto, os problemas relacionados com o *buffering* podem ser ultrapassados criando filas separadas para áudio e vídeo nos *switches* e *routers* de rede. Assim, a existência de filas separadas permite que os pacotes sensíveis ao atraso, como áudio e vídeo, possam ser transmitidos de forma prioritária. De referir, ainda, que os *hubs* de rede não suportam filas de dados, o que pode levar ao aumento de colisões, causando perdas de pacotes ou atrasos. Por este motivo, os *switches* têm preferência em relação aos *hubs* numa rede desenhada para apresentar *Quality of Service*. Porém, nem todos os *switches* de rede suportam filas separadas ou esquemas de classificação, sendo que esses casos têm de ser atualizados ao implementar o *Quality of Service* [16].

2.2.3 *Power Over Ethernet*

A tecnologia *Power over Ethernet* (PoE) permite a transmissão de energia elétrica juntamente com os dados para um dispositivo remoto, através do cabo de rede *Ethernet* [17]. Esta tecnologia é útil para fornecer energia pelo cabo de rede a telefones IP, *access points*, câmaras IP, *switches* remotos, *embedded devices*, bem como a outros equipamentos aos quais não seja possível fornecer energia pela rede elétrica, conforme ilustrado na figura 2.5.

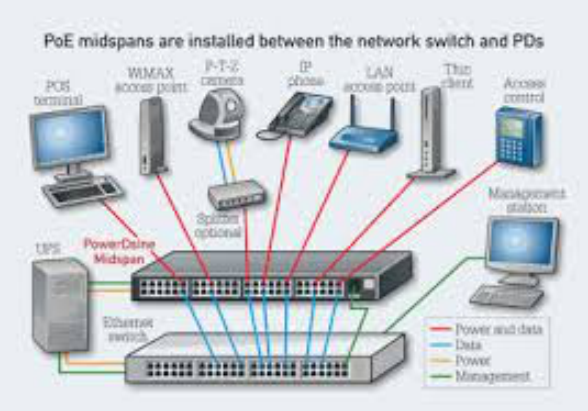


Figura 2.5: *Power over Ethernet* (extraído de [18]).

O *standard* IEEE 802.3af regulamenta todo o conceito de PoE, desde 2003, e deter-

mina que a alimentação deve ser de 48 Vcc e a potência do sinal deve ser, no máximo, de 15,4 *watts*. A ratificação do *standard* IEEE 802.3at (PoE+), em 2009, aumentou a potência disponível em cada porta, permitindo o desenvolvimento de novas categorias de dispositivos capazes de utilizar PoE, incluindo *multi-band wireless access points*, *video phones* (VoIP) e monitorização de câmaras IP com capacidade de controlar o ângulo da mesma, entre outros. Este standard determina que a alimentação deve ser de 48 Vcc e a potência do sinal deve ser, no máximo, de 34,2 *watts*. De referir que o *standard* IEEE802.3bt, que a esta data se encontra em processo de revisão (*draft*), pretende aumentar ainda mais a potência disponível [17].

O *Power over Ethernet* é uma forma conveniente e segura para fornecer energia aos dispositivos. Ao injetar corrente nos cabos de rede, já não é preciso ter uma tomada de alimentação no local onde se encontra o equipamento. Importa também referir outras vantagens, tais como a redução da quantidade de cabos necessários, o facto de ser uma forma central de fornecer energia para facilitar o uso de UPS e, dado que não é necessário instalar novas tomadas, apresenta um baixo custo [19]. A principal desvantagem do *Power over Ethernet* é o facto de estar limitado na potência fornecida, não permitindo alimentar certos aparelhos a partir do cabo de rede, no entanto, a cada *standard* publicado, a potência fornecida tem vindo a aumentar [19].

2.2.4 *Power Line Communication*

A tecnologia *Power Line Communication* (PLC) permite aos consumidores conectar aplicações de domótica entre si, através do sistema elétrico instalado [20]. Esta tecnologia consiste na transmição de dados e voz pela rede de energia elétrica e, tendo em conta que utiliza a cablagem existente, não necessita de infraestruturas adicionais, conforme ilustrado na figura 2.6.

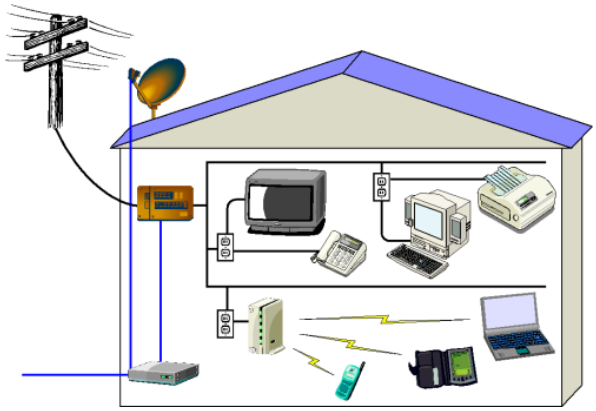


Figura 2.6: *Power Line Communication* (extraído de [21]).

Os protocolos *standard* HomePlug e CENELEC são os mais utilizados tanto para débitos de transferência baixos como elevados [20]. Esta tecnologia trabalha na camada 2 do modelo *Open Systems Interconnection* (OSI), pelo que pode ser agregada a uma rede IP (camada 3) já existente, além de ter capacidade para trabalhar em conjunto com outras tecnologias da camada 2. O *modem* PLC utiliza a rede elétrica para fazer todas

as transmissões, sendo que as camadas utilizadas do modelo OSI são a camada 1 e a camada 2.

O princípio básico de funcionamento das redes PLC consiste no facto de, por um lado, a frequência dos sinais de conexão ir, no caso de *narrowband*, de 3 a 500 KHz e, no caso de *broadband*, de 1,8 a 250 MHz e, por outro, a energia elétrica possuir uma frequência de 50 a 60 Hz, o que permite aos dois sinais conviverem, no mesmo meio, sem interferências. A tecnologia PLC apresenta velocidade simétrica, ou seja, tem o mesmo desempenho na receção e no envio de dados [22]. As principais aplicações usadas em PLC são a leitura automática de medidores, as redes domésticas e de acesso à internet, a domótica, a transmissão de programas de rádio, o controlo de luz, as redes de comunicação de débitos baixos, a gestão de distribuição de potência e os edifícios inteligentes, entre outros [20] [23].

Muitas das habitações não têm cablagem dedicada para rede de dados e o custo de instalação da mesma é, normalmente, elevado. A *Power Line Communication* usa a rede elétrica existente para a comunicação, assim o custo de instalação é mais baixo, quando comparado com outros sistemas de comunicação, e o acesso está disponível em qualquer tomada de rede. Se existirem várias tomadas nas diversas divisões, a rede elétrica representa uma excelente rede para partilhar dados entre os vários dispositivos, com altos débitos de transferência. No entanto, a *Power Line Communication* não é a melhor solução para manter os dados seguros. Adicionalmente, o ruído presente na linha elétrica pode provocar atenuação no sinal e limitar os débitos possíveis de serem praticados [20].

2.2.5 Convergência Tecnológica

O Livro Verde sobre Convergência define a convergência como a capacidade de diferentes plataformas de rede transportarem, essencialmente, serviços semelhantes ou dispositivos do consumidor, tais como o telefone, a televisão e o computador pessoal [24]. A convergência ocorre a vários níveis, sendo de destacar as plataformas tecnológicas e de rede, as alianças industriais e fusões, os serviços e mercados e, ainda, a política e regulação [24]. Deste modo, a definição de convergência implica várias dimensões para o mesmo conceito como, por exemplo, a convergência da indústria, a convergência de serviços e a convergência de rede, entre outros, sendo que o pressuposto básico é que a convergência tecnológica conduz a novos processos de convergência.

A convergência tecnológica consiste na utilização de uma única infraestrutura para fornecer serviços que, anteriormente, requeriam equipamentos, canais de comunicação, protocolos e padrões independentes. A convergência tecnológica permite que o utilizador aceda às informações a partir de qualquer lugar e através de qualquer meio de comunicação, por uma interface única, com impacto positivo em vários setores, nomeadamente na economia, na comunicação e na produção, entre outros.

Os principais fatores de convergência têm sido identificados na literatura existente, sendo que o *driver* mais importante do processo é a mudança tecnológica juntamente com a liberalização dos mercados de telecomunicações [24]. De salientar também o importante papel da regulação, uma vez que tem permitido que o processo de convergência avance em diferentes áreas da economia, e dos fatores socioeconómicos, incluindo

as opiniões e reações dos utilizadores, bem como da *internet* e disponibilidade de dados e informações [24]. Desta forma, a convergência tecnológica resulta da mudança tecnológica e do surgimento de novas tecnologias.

Este avanço tecnológico tornou possível a interoperabilidade de sistemas, o desenvolvimento de novos dispositivos facilitadores da mobilidade e interatividade e, ainda, a obtenção de serviços integrados que disponibilizam mais informação e serviços.

Por último, importa fazer referência ao conceito de computação em nuvem (*cloud computing*) que consiste na utilização da memória e das capacidades de armazenamento e cálculo de computadores e servidores compartilhados e interligados por meio da *internet* [25].

2.2.6 Computação em Nuvem

A computação em nuvem é uma forma moderna de arquitetura e implementação de serviços com base em mudanças evolutivas, inicialmente, utilizada apenas em laboratórios pela *Amazon*, *Google* e *Microsoft*, que os especialistas consideram uma nova fronteira da era digital. A computação em nuvem permite que, a partir de qualquer computador e em qualquer lugar do mundo, seja possível ter acesso a informações, arquivos e programas num sistema único, independente da plataforma utilizada, sendo que o acesso a programas, serviços e arquivos é remoto, conforme apresentado na figura 2.7.

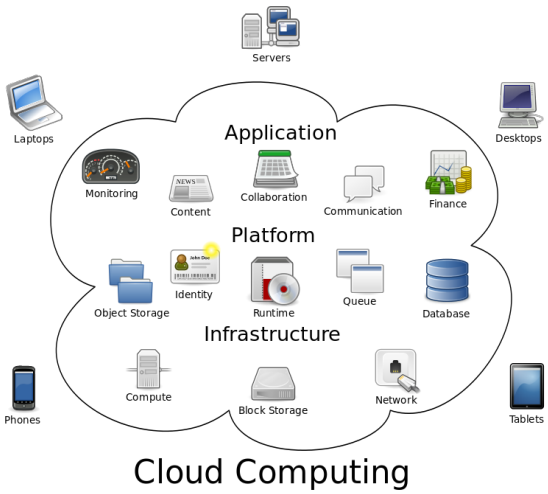


Figura 2.7: *Cloud Computing* (extraído de [26]).

Caraterísticas

A computação em nuvem apresenta duas caraterísticas essenciais [25]:

- Abstração: a computação em nuvem abstrai os detalhes da implementação do sistema dos utilizadores e programadores, sendo que as aplicações correm em sistemas físicos que não são especificados, os dados são armazenados em locais que não

são conhecidos, a administração de sistemas é efetuada em *outsourcing* e o acesso dos utilizadores é generalizado; e,

- Virtualização: a computação em nuvem virtualiza sistemas na medida em que reúne e partilha recursos, ou seja, os sistemas e o armazenamento podem ser providenciados sempre que necessário, a partir de uma infraestrutura centralizada, os custos são avaliados com base no uso, permite a utilização de software *multitenancy* e os recursos são escaláveis de forma rápida e eficaz.

Adicionalmente, existe uma terceira caraterística da computação em nuvem, que está relacionada com o facto de a nuvem ser vista como uma *utility* e os serviços prestados com base num modelo *pay-as-you-go*.

Tipologia da Nuvem

Importa distinguir duas classes de nuvens: as baseadas no modelo de implementação e as baseadas no modelo de serviço.

Modelo de Implementação

O modelo de implementação apresenta a localização da nuvem e qual a sua finalidade. Nos modelos de implementação, importa referir as nuvens públicas, privadas, comunitárias e híbridas. A definição de NIST para os quatro modelos de implementação é a seguinte [25]:

- Públicas: a infraestrutura das nuvens públicas é de utilização universal. A nuvem pública toma em consideração questões fundamentais como o desempenho e a segurança, pelo que as aplicações nela executadas mantêm-se acessíveis, tanto para os prestadores de serviços como para os utilizadores.
- Privadas: as nuvens têm um único utilizador que, sendo proprietário da infraestrutura, detém total controlo sobre as aplicações nela implementadas.
- Comunitárias: a infraestrutura da nuvem é partilhada por diversos utilizadores que têm um propósito comum.
- Híbridas: integram nuvens públicas e privadas, permitindo que estas utilizem recursos de nuvens públicas, com a vantagem de manter os níveis de serviço mesmo que haja a necessidade de aumentar os recursos utilizados.

Modelo de Serviço

O modelo de serviço descreve o tipo de serviço que o prestador de serviços oferece. Os modelos de serviço mais conhecidos são o *Software as a Service* (SaaS), quando se usa um *software* em regime de utilização *web*, o *Platform as a Service* (PaaS), quando se utiliza apenas uma plataforma como, por exemplo, um banco de dados ou um *webservice*, e o *Infrastructure as a Service* (IaaS), quando se utiliza uma percentagem de um servidor [25].

Existem, no entanto, outros modelos de serviço tais como o *Storage as a Service* (StaaS), o *Identity as a Service* (IdaaS) e o *Compliance as a Service* (CmaaS), entre outros.

Mudança de Paradigma

A computação em nuvem representa uma mudança de paradigma real na forma como os sistemas são implementados. A escala massiva de sistemas de computação em nuvem foi possibilitada pela popularização da *internet* e o crescimento de algumas grandes empresas de serviços. Os prestadores de serviços de computação em nuvem disponibilizam grandes infraestruturas de *datacenters*, computadores, capacidade de armazenamento e *networking*. Para perceber a forma como a computação em nuvem mudou a natureza da implementação do sistema comercial, importa referir os seguintes exemplos [25]:

- *Google*: construiu uma rede mundial de centros de dados para suportar o seu motor de busca. Neste processo, a *Google* capturou uma parte substancial da receita de publicidade do mundo, que permitiu oferecer *software* gratuito aos utilizadores e mudou o mercado de *software* orientado para o utilizador.
- Plataforma *Azure*: a *Microsoft* está a criar a plataforma *Azure*, permitindo que os aplicativos *.NET Framework* sejam executados através da *internet* como uma plataforma alternativa para os programadores de *software* em execução em *desktops*.
- *Amazon Web Services*: um dos negócios baseados em nuvem mais bem sucedidos, que consiste num modelo de serviço *Infrastructure as a Service*, permitindo alugar computadores virtuais em infraestrutura própria da *Amazon*.

Com o constante crescimento verificado nas últimas décadas, as empresas desenvolveram os seus *datacenters* como projetos *greenfield*, o que permitiu tornar as redes de computação em nuvem altamente eficientes e capturar margem suficiente para rentabilizar o *utility computing* [25].

Vantagens e Desvantagens

A grande vantagem da computação em nuvem é a possibilidade de utilizar *softwares* sem que estes estejam instalados no computador. Adicionalmente, a computação em nuvem apresenta outras vantagens como, por exemplo, o facto de o cliente poder aprovisionar recursos computacionais de forma automatizada, sem a necessidade de interação humana e permitir o acesso a recursos que estão disponíveis na *cloud*, através de métodos *standard* de forma a que seja possível aceder a esses conteúdos a partir de qualquer aparelho. Para além destas vantagens, é também de mencionar outros aspetos positivos, tais como ser barato, de fácil utilização, com qualidade de serviço e fiabilidade, permitir colocar a gestão IT em *outsource*, simplificar a manutenção e *upgrade* dos sistemas bem como apresentar uma barreira de entrada baixa, permitindo investir num serviço deste gênero, uma vez que as despesas diminuem consideravelmente [25].

A maior desvantagem da computação em nuvem é a falta de acesso à *internet*. Caso falhe o acesso, ficam comprometidos todos os sistemas. No entanto, também importa referir que os serviços *cloud* podem não ser tão costumizáveis quanto se pretende e as aplicações que correm na *cloud* têm um problema intrínseco associado à latência da conexão WAN. Para que a conexão possa sobreviver a um sistema distribuído, as comunicações são por natureza, necessariamente, unidirecionais. Por fim, importa referir que os aspetos da segurança e privacidade são a maior área de preocupação [25].

Os sete princípios de segurança numa rede em nuvem são os seguintes: acesso privilegiado de utilizadores, *compliance* com regulamentação, localização dos dados, segurança dos dados, recuperação de dados, apoio à investigação e viabilidade a longo prazo [27].

No capítulo seguinte apresenta-se a arquitetura do sistema *Integrated Access Point*, evidenciando os programas desenvolvidos e a interação do utilizador com este sistema.

Capítulo 3

Arquitetura do Sistema

Neste capítulo apresentam-se as especificações do sistema desenvolvido no âmbito desta dissertação, que se baseia na extensão de um *Access Point* 802.11 (com interface *Fast/Gigabit Ethernet*) por forma a integrar dispositivos e mecanismos de suporte à domótica e sua conexão à rede de dados, conforme apresentado na figura 3.1.

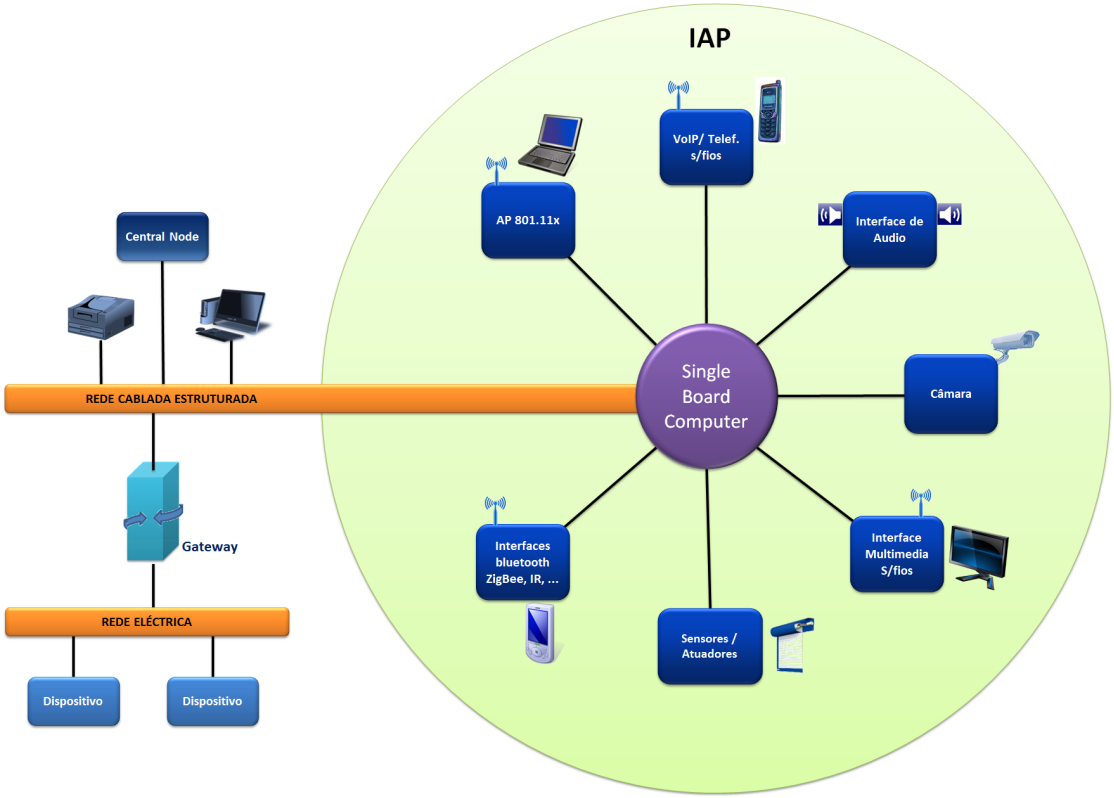


Figura 3.1: Sistema *Integrated Access Point*.

Assim, as funções do *Integrated Access Point* (IAP) para sistemas domóticos, com utilização combinada de redes cabladas ou não cabladas (ambas *standard* de uso geral), são:

- Permitir, a montante, a utilização de uma rede de dados cablada;
- Disponibilizar, a jusante, uma rede de dados não cablada; e,
- Integrar no *Center* os componentes e mecanismos para disponibilizar os serviços adicionais, assim como as respetivas interfaces para acesso à rede.

A escolha de um equipamento do tipo *access point* para integrar funcionalidades de aplicações domóticas justifica-se pelo facto de se acreditar que, num futuro próximo, para garantir uma boa cobertura e uma elevada largura de banda, é necessária a instalação de múltiplos *access points* numa habitação. No limite, pode existir um *access point* por divisão, assegurando a conectividade de todos os dispositivos aí existentes, sendo que todos estarão interligados por uma rede cablada estruturada de alto débito.

A modularidade do IAP, quer ao nível do *hardware*, quer do *software*, é um aspeto fundamental para permitir a seleção "*à la carte*" das funcionalidades pretendidas.

Considerando que na generalidade dos estudos e elaboração de projetos de domótica se pretende a integração de diferentes tipos de serviços num *access point*, de modo a permitir o envio e receção sobre a mesma rede cablada de diferentes tipos de tráfego (dados genéricos e informação específica relacionada com as funcionalidades adicionais implementadas), a comunicação deve ser feita em ambos os casos com garantias de qualidade de serviço (por exemplo, largura de banda mínima e latência máxima).

Por último, pretende-se privilegiar a utilização de pilhas protocolares de comunicação do tipo *TCP/IP*, por forma a simplificar a comunicação entre dispositivos e a integração com a *internet*.

3.1 Funcionalidades

Este sistema tem como propósito interligar todos os sistemas domóticos, usando para o efeito os *access points* distribuídos por uma habitação ou edifício. A ideia passa por ter um *access point* em cada divisão, de modo a que todos os sistemas que se encontrem nessa divisão se possam ligar a esse *access point*, por exemplo, as persianas, as luzes, os interrutores e o ar condicionado. Uma vez que todos os *access points* estão ligados entre si, tendo um servidor que funciona como ponto central, em torno do qual são geridos todos os serviços que correm em todos os *access points*, e sendo disponibilizada uma interface gráfica para o utilizador poder interagir com esses mesmo serviços, torna-se possível usufruir da rede cablada já existente e que é utilizada para uso geral de dados, através da adição de funcionalidades. Assim, o *access point* deixa de ter apenas a função de providenciar acesso *wireless* à rede, mas também de interligar todos os dispositivos de domótica existentes nessa divisão. Com este sistema, é possível fazer adições posteriores à instalação, na medida em que basta adicionar novos módulos, quer seja em componente de *hardware*, quer seja em componente de *software*.

Como existem vários *access points* numa habitação, e em cada um deles podem correr vários serviços, é necessário criar uma infraestrutura que permita fazer uma gestão eficiente dos recursos, bem como a devida identificação de cada serviço em cada *access point*. Para este efeito, criou-se um conjunto de programas que são instalados tanto nos

access points como no servidor que serve de centro de comandos, de forma a manter a integridade da informação transmitida entre os diferentes nodos.

Assim, o utilizador pode interagir com o sistema a partir de um *browser*, o que permite que este possa usar um computador, um *smartphone*, um *tablet* ou uma *smart TV*. Além disso, o sistema possui um servidor VPN, instalado como módulo, que permite ao utilizador, mesmo não estando dentro da rede, isto é, mesmo não estando dentro da habitação ou edifício, poder interagir com o sistema, permitindo o controlo à distância de forma segura e privada.

3.2 Arquitetura da Infraestrutura da Rede

Nesta secção aborda-se a infraestrutura da rede tecnológica, que é composta pelo núcleo do IAP (*Central Node*) e interfaces aos dispositivos (APs), apresentando-se uma visão global do funcionamento da rede, que pode ser acedida a partir de qualquer dispositivo, desde que disponha de um *browser* para permitir a gestão da infraestrutura criada.

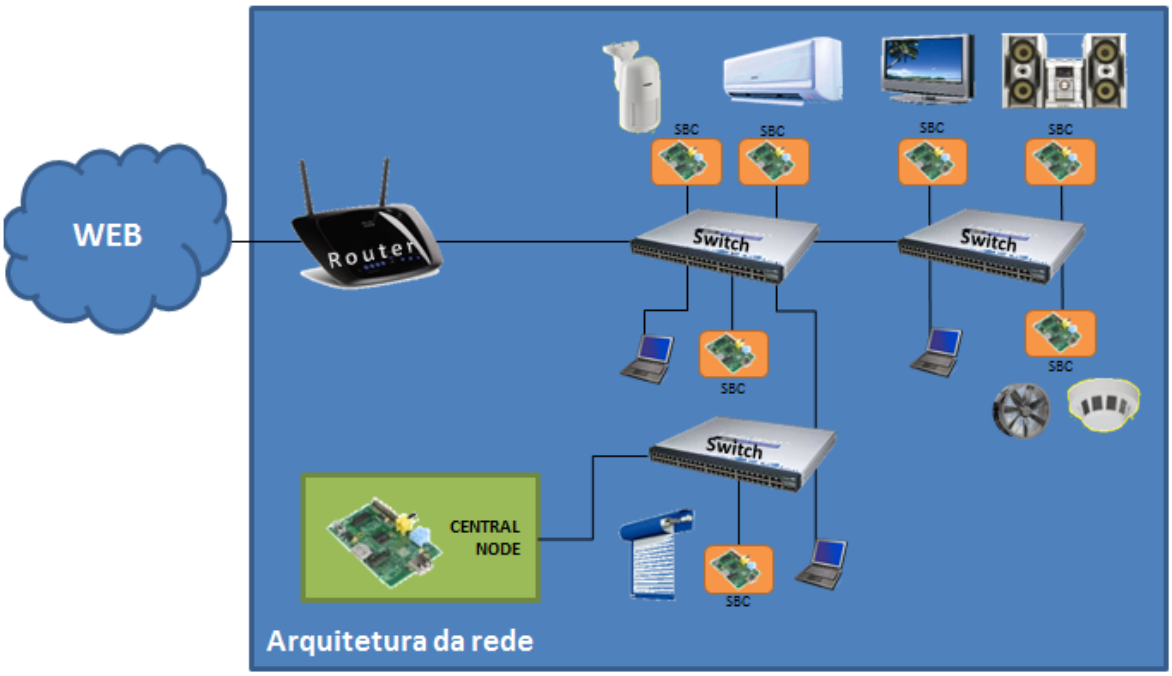


Figura 3.2: Arquitetura do Sistema IAP.

Na figura 3.2 apresenta-se um exemplo de uma rede que utiliza o sistema IAP. Cada *Single Board Computer* representa um *access point*, ao qual se pode ligar as persianas elétricas, as luzes, os interrutores, sensores de fogo e sensores de qualidade de ar, entre outros. Para além dos *Single Board Computers* que funcionam como *access points*, é necessário ter um *Single Board Computer* que funcione como centro de comandos (*Central Node*). Este último é responsável por comunicar com todos os outros *access points*, bem como coletar toda a informação enviada pelos mesmos. Para além disso, tem a função de

disponibilizar uma página *web* ao utilizador, a partir da qual se pode fazer toda a gestão da rede como, por exemplo, monitorizar vários valores e executar ordens, tais como baixar a persiana ou apagar a luz de uma divisão.

Nas figuras 3.3 e 3.4 apresentam-se as componentes da infraestrutura da rede tecnológica: núcleo do IAP (*Central Node*) e interfaces aos dispositivos (*access points*).

3.2.1 *Central Node*

O núcleo do IAP, designado por *Central Node*, é o centro de comandos de todo o sistema e incorpora três programas, desenvolvidos no âmbito deste trabalho, cujos conceitos, objetivos e funções são definidos mais em pormenor na secção seguinte:

- Dois programas de infraestrutura da rede: *discovery center* (DC) e *intAPcomConv* (IAPcC); e,
- Uma página *web* para gestão da rede, disponibilizada através de um *web server* (*Apache*).

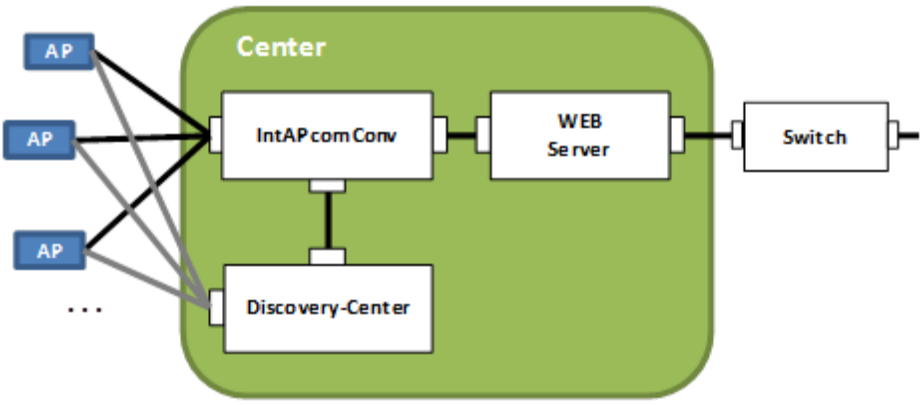


Figura 3.3: Programas Desenvolvidos para o Núcleo *Central Node* do Sistema IAP.

De notar que o *Central Node* necessita de interfaces aos dispositivos *access points*, para estabelecer a comunicação com os equipamentos, cuja representação se pode observar na figura 3.3.

3.2.2 *Access Points*

As interfaces aos dispositivos, designados por APs, são periféricas do *Central Node*, fundamentais para estabelecer a ligação aos equipamentos da habitação que se pretende controlar, e incorporam três programas desenvolvidos, cujos conceitos, objetivos e funções são definidos de forma mais pormenorizada na secção seguinte:

- Dois programas de infraestrutura da rede: *intAPcom* (IAPc) e *discovery notifier* (DN); e,

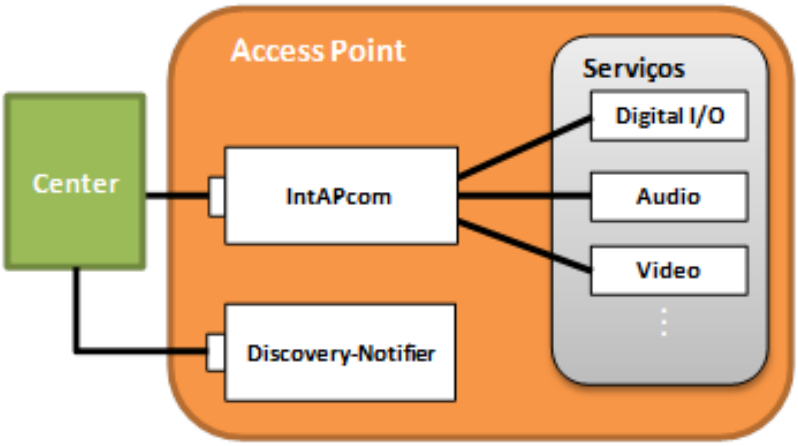


Figura 3.4: Programas Desenvolvidos para a Interface (APs) do Sistema IAP.

- Um programa que opera serviços de *Digital I/O* (DIO): *Shutter Commander*.

De notar que, como exemplo de serviços que podem ser integrados no *access point*, apenas foi desenvolvido o programa *Shutter Commander*, que permite controlar uma persiana por forma a validar todo o funcionamento da infraestrutura. No entanto, qualquer outro programa pode ser desenvolvido acoplado a esta infraestrutura.

Com esta infraestrutura da rede tecnológica, o utilizador pode comunicar com o sistema utilizando uma interface *web*, conforme apresentado na secção 3.4.

3.3 Aplicações e Serviços de Rede

Todos os programas foram desenvolvidos com o objetivo de criar uma infraestrutura da rede de suporte a programas especializados, que operam serviços no interior da habitação, permitindo a interação de dispositivos muito diferenciados, com necessidades de largura de banda e latência muito distintas.

O conjunto de programas desenvolvidos para a infraestrutura de rede foram o *discovery notifier* (DN), o *discovery center* (DC), o *intAPcom* (IAPc), o *intAPcomConv* (IAPcC) e a *web page*, tendo também sido concebido, como exemplo de serviço, o *digital I/O*: *Shutter commander* (Sc). Este conjunto de programas constitui o sistema da arquitetura da rede que se designa por *Integrated Access Point (IAP)*.

Na figura 3.5 apresenta-se o *overview* da representação esquemática do sistema da arquitetura da rede (*hardware e software*) desenvolvida.

Apresentada a conceção da arquiretura da rede, importa definir os conceitos de cada programa, os objetivos, o local onde são instalados os APs e as funções para a infraestrutura da rede e para os programas que operam serviços.

3.3.1 Programas que Constituem a Infraestrutura da Rede

Foram desenvolvidos quatro programas para a infraestrutura da rede, bem como uma *web page*, a seguir apresentados:

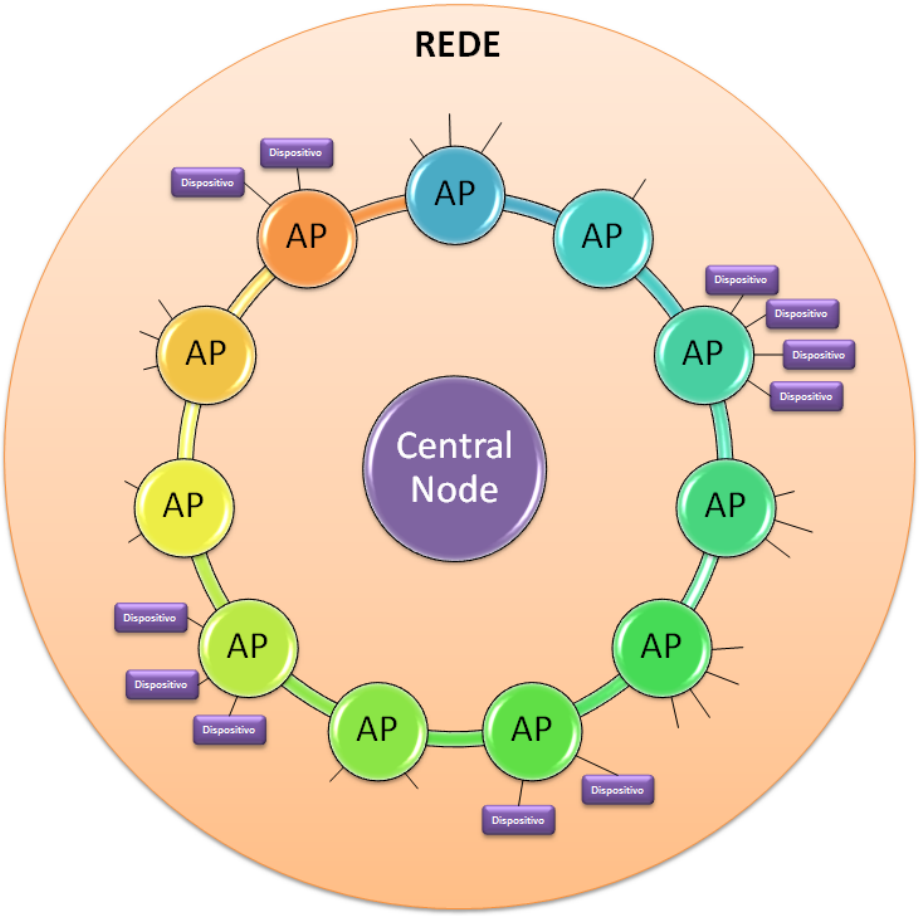


Figura 3.5: *Overview* da Arquitetura da Rede.

- *Discovery suite* (DS): A *discovery suite* é composta por dois programas, *discovery notifier* e *discovery center*.
  - O *discovery notifier (DN)* é um programa que envia pacotes de informação em modo *Broadcast* para a rede, com o objetivo de se identificar através de *host-name*, *IP Address* e *MAC Address*, encontrando-se instalado em todos os APs da rede, com a função de informar que se encontra *online*. Esta mensagem de identificação é emitida em intervalos de tempo, sendo configurável em função da importância e urgência em saber que os APs estão *online* ou *offline*.
  - O *discovery center (DC)* é um programa que recebe, exclusivamente, pacotes de informação do DN, dispondo de duas portas (uma para receber informação do DN e outra para dialogar com o IAPcC), com o objetivo de "escutar" todas as mensagens em circulação na rede, encontrando-se instalado no *Central Node*, com a função de manter uma tabela atualizada com a identificação dos APs e do seu estado *on/off*. Ao fim do tempo pré-configurado, se o *discovery center* não receber comunicação de um qualquer AP identificado na tabela de *hosts*, de imediato é colocada uma anotação informando que o AP está *offline*.

- *IntAPcom* (IAPc): Este programa recebe, do IAPcC, pacotes de informação e o ID do serviço a que se destina, transmite-os aos dispositivos e devolve a comunicação informando o seu estado (por exemplo, não foi possível executar), com o objetivo de mediar o cumprimento de ordens de serviço emanadas do *Central Node* e os serviços instalados (por exemplo, estores, áudio e vídeo).

  O *intAPcom* encontra-se instalado em todos os APs da rede, com a função de fazer a gestão (*routing*) das mensagens dos serviços instalados em cada um deles.

  Em resumo, o *intAPcom* recebe ordens de serviço e identificação do serviço a que se destina, emanadas pelo programa *intAPcomConv*, retransmite a ordem ao respetivo serviço (por exemplo, abertura parcial de estores) e devolve a informação ao *intAPcomConv* nos seguintes termos: executou ou falhou.

- *IntAPcomConv* (IAPcC): Este programa é acionado quando o utilizador interage com a rede IAP, via *web page*, de outro modo fica *idle*. Com o objetivo de expedir pacotes com a ordem e o ID do serviço, tem de se conectar com o DC para fazer o *request* do *IP address* dos APs envolvidos que estejam *online*.

  O *intAPcomConv* encontra-se instalado no *Central Node* com a função de, recebido o pacote de informação do utilizador da rede, através da *web page*, com a ordem de serviço, o *ID Service* e o *IP address*, enviá-lo ao IAPc, eliminando o *IP address* do pacote.

- *Web Server* (WS): O servidor web (*Apache*) encontra-se instalado no *Central Node*, tendo a função de permitir ao utilizador da rede IAP obter informações sobre o funcionamento e estado da rede, dar instruções (por exemplo, baixar o estore) e receber notificações de eventos ocorridos.

3.3.2 Programas que Operam Serviços

No âmbito do presente trabalho foi desenvolvido um serviço *digital I/O*, para demonstração da infraestrutura, tendo sido materializado num exemplo concreto designado por *Shutter Commander*, relacionado com a monitorização de estores, conforme se apresenta de seguida:

- *Digital I/O* (Dio): Os serviços baseados em *digital I/O* podem ser comandados através de ordens simples, tais como a ordem de *on* ou *off*, sendo serviços cujas ordens são do tipo booleanas, ligar/desligar, subir/descer, mover/parar, ou ordens baseadas numa escala, tais como subir 10% ou escurecer 50%. Estes serviços contemplam uma diversidade de funcionalidades como, por exemplo, controlar lâmpadas, toldes, estores, alarmes ou atuadores, podendo ainda servir para ler dados de sensores.

  O *Shutter Commander* é um serviço que permite enviar ordens (via rede) ao dispositivo de comando de estores para que este as execute, sejam elas, subir, descer ou colocar o estore num determinado ponto, encontrando-se instalado nos APs da rede, com a função de controlar os estores de uma habitação. Através deste serviço é possível saber quando o estore está em movimento, seja para cima ou para baixo,

ou até no ponto do percurso em que se encontra, sendo que, para isso, basta estar ligado à página *web*.

Para além do programa desenvolvido, muitos outros podem ser concebidos, nomeadamente ao nível do áudio e vídeo, conforme se descreve abaixo:

- O serviço de áudio permite disponibilizar som numa ou mais divisões da habitação, a partir de um emissor de áudio, com o objetivo de ser possível a partir de fontes diferentes de emissão de áudio, distribuí-lo para outros locais da habitação, encontrando-se instalado nos APs da rede e com a função de rotear de acordo com a informação recebida os pacotes de áudio para os APs pretendidos. Este sistema é desenvolvido no pressuposto que qualquer AP pode ser uma fonte de emissão, para além de ser recetor de áudio, possível de ser configurável através da *web page*.

- O serviço de vídeo permite disponibilizar um serviço semelhante ao de áudio, ou seja, permite disponibilizar vídeo numa ou mais divisões da habitação a partir de um emissor de vídeo, tendo por base a lógica de funcionamento descrito para o áudio. Neste tipo de serviço pretende-se aproveitar o potencial disponível, alargando a conexão às câmaras de vigilância, vídeo porteiro e *baby-monitoring*, entre outros.

3.3.3 Comunicação na Infraestrutura da Rede

A comunicação na infraestrutura da rede passa por vários programas, sendo que informação vai sendo adicionada ou retirada aos pacotes, consoante a comunicação se efetua do cliente para o serviço ou do serviço para o cliente.

Quando o utilizador envia um comando para um serviço, a partir da página *web*, esse comando passa por várias etapas. A mensagem é enviada pelo *browser* ao servidor de *websockets* (*intAPcomConv*). Esta mensagem é composta pelo IP do IAP e pelo serviço para o qual se pretende enviar o comando, bem como pelo próprio comando. Quando o pacote chega ao servidor de *websockets* (*intAPcomConv*), é analisado o IP para o qual se pretende reencaminhar o pacote. De seguida, retira-se o parâmetro IP do pacote e envia-se para o devido IAP. Quando a mensagem chega ao distribuidor de mensagens local (*intAPcom*) que corre no IAP, o parâmetro que identifica o serviço é verificado para se poder dar o seguimento devido ao pacote. Depois é retirado o comando do pacote e enviado para o serviço em questão.

Quando um serviço pretende notificar algum utilizador sobre um determinado evento, envia a notificação para o distribuidor de mensagens instalado no IAP onde o serviço está a correr. O distribuidor de mensagens cria um pacote, onde introduz a notificação e o nome do serviço de onde esta provém e envia para o *intAPcomConv*. Este, por sua vez, adiciona ao pacote o IP do IAP de onde proveio a notificação, enviando de seguida o pacote para o *browser*, através de *websockets*. O *browser* recebe o pacote, analisa os parâmetros e mostra ao utilizador a notificação, bem como o serviço e o IP de onde foi originada.

O browser pode mostrar as notificações de várias formas, desde que, para o efeito, sejam criadas as respetivas funções. Por exemplo, se existirem vários estores ligados a

diferentes IAPs, é possível mostrar na página *web* a informação agregada de todos os estores, realçando mudanças de estado, em caso de receção de uma notificação.

3.4 Interface do Utilizador com a Tecnologia

Na secção 3.2 apresentou-se a infraestrutura da rede tecnológica, composta pelos seguintes elementos:

- Núcleo do IAP, *Central Node*, que é o centro de comandos do sistema IAP;
- Interfaces aos dispositivos, *access points*, que são elementos fundamentais para conectar o *Central Node* aos equipamentos da habitação; e,
- *Router*, que é o elemento essencial de ligação à *web*, para permitir aceder ao sistema IAP a partir do exterior.

Nas duas secções seguintes, *Interface Web* e Navegação por Menus, aborda-se exclusivamente a ligação ao *web server* e demonstra-se a facilidade com que se gere o sistema IAP.

3.4.1 Interface *Web*

A interface *web* permite ao utilizador da habitação aceder a uma *web page*, a partir de qualquer dispositivo que utilize um *browser*, ultrapassando-se barreiras de desenvolvimento de *software* para diferentes plataformas.

3.4.2 Navegação por Menus

A navegação por menus, *user friendly*, consiste num processo muito simples de rapidamente identificar, com base na *tab* principal que tem a lista dos serviços disponíveis na rede, a seguinte informação: *on/off*, *hostname*, *IP Address* e *MAC Address*.

A *web page* foi criada com ajuda da tecnologia *websockets*, o que permite troca de informação com os diversos APs em tempo real. Sem esta tecnologia seria necessário fazer *polling* ao *webserver* à procura de nova informação. Com os *websockets*, o *webserver* pode alertar o utilizador da ocorrência de algum evento sem que este tenha de a requisitar constantemente.

Para além da *tab* principal, existem tantas tabs secundárias quantos os serviços que estiverem em funcionamento, ou seja, que tiverem sido instalados. Estas tabs permitem ao utilizador monitorizar, em tempo real, o estado da sua instalação, selecionando os serviços e controlando ou avaliando os diferentes dispositivos da sua habitação.

No decurso desta dissertação desenvolveu-se a *tab* principal, uma *dummy service* (para efeitos de teste e *debugging*) e uma *tab* secundária, cujos *layouts* se apresentam abaixo.

Tab Principal

A *tab* principal designa-se de *APs List* e possui a configuração que consta da figura 3.6.

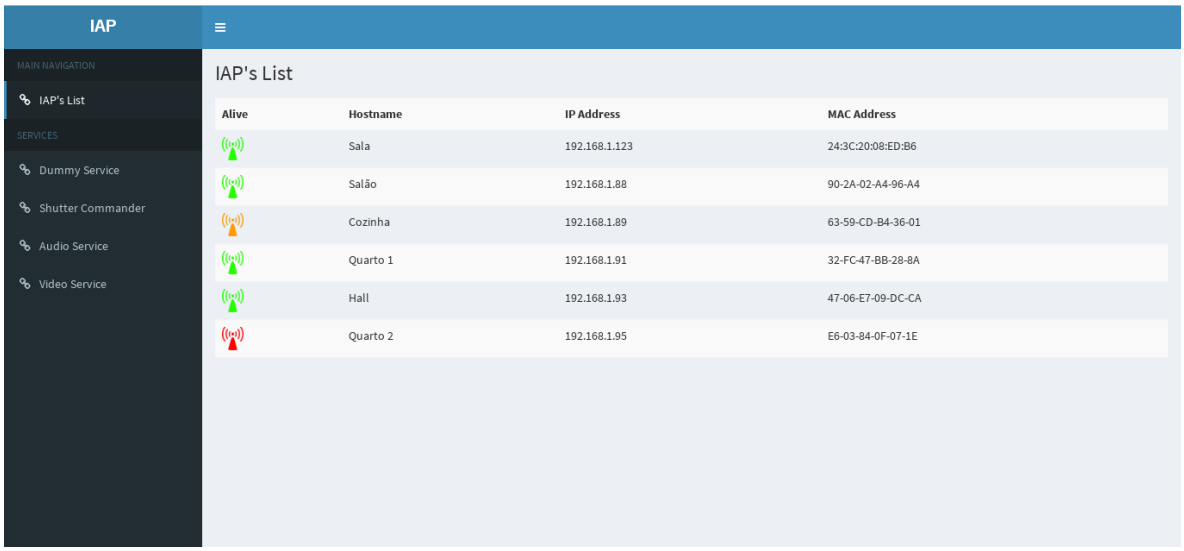


Figura 3.6: *APs List Web Page*.

Como se pode constatar pelo *layout*, a primeira informação é a indicação de que o AP está ou não ativo (*alive*), seguindo-se a identificação do serviço (*hostname*), identificação do AP (*IP Address*) e endereço físico (único) do equipamento (*MAC Address*).

Dummy Service

O *dummy service* é utilizado apenas para realizar testes de funcionamento à instalação da rede e utiliza-se sempre que, num novo projeto, se prepara a instalação.

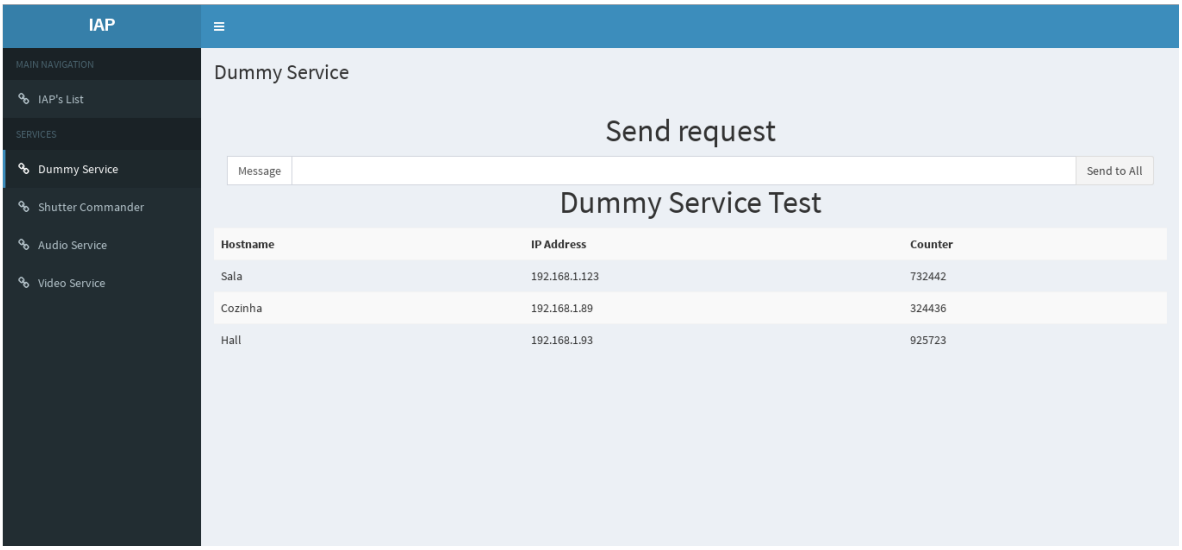


Figura 3.7: *Dummy Service Web Page*.

Resumidamente, como a figura 3.7 demonstra, são apresentados dois itens e controlado um item, *hostname*, *IP Address* e *counter*, sendo que este último consiste num número aleatório incrementado pelo AP e enviado ao *Central Node*, garantindo-se, deste

modo, que a comunicação se encontra ativa (*online*).

Tabs Secundárias

As *tabs* secundárias serão tantas quantos os serviços utilizados, tendo sido desenvolvido no âmbito da presente dissertação, o *layout* relativo ao serviço de estores (*Shutter Commander*), conforme apresentado na figura 3.8.

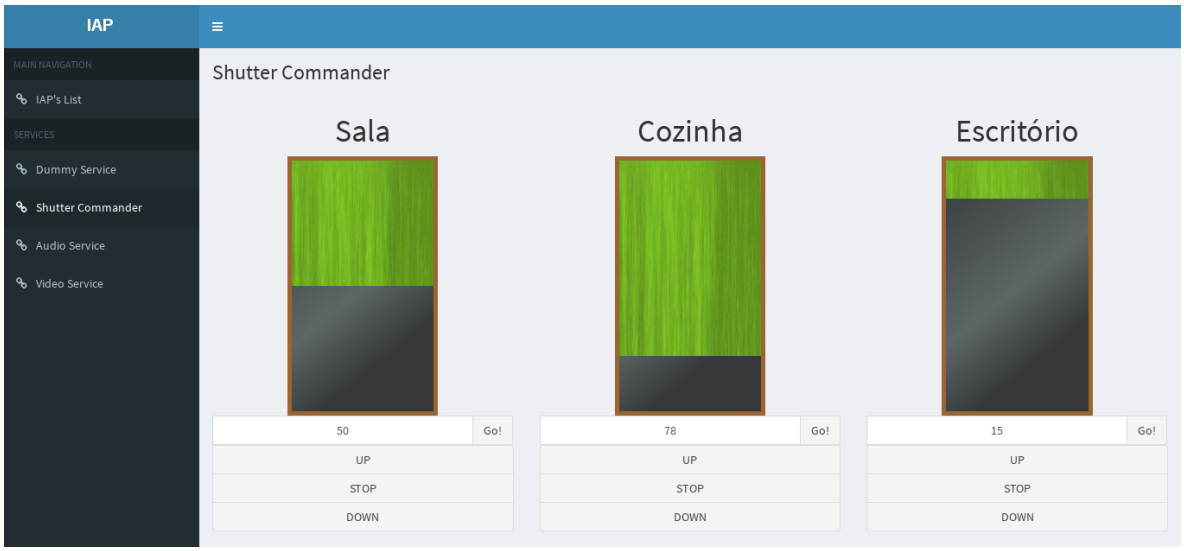


Figura 3.8: *Shutter Commander Web Page*.

Como se pode constatar pelo *layout*, é possível enviar ordens de *UP*, *DOWN* ou *STOP*, para subir, descer ou parar o estore, respetivamente, bem como especificar um ponto de paragem. Para além disso o estado do estore é atualizado em tempo real.

No capítulo seguinte aborda-se as metodologias utilizadas no desenvolvimento dos programas e detalha-se o funcionamento de cada um dos programas.

Capítulo 4

Implementação do Sistema

Neste capítulo pretende-se demonstrar a abordagem usada para o desenvolvimento dos vários programas e, ainda, ilustrar a planificação, documentação e elaboração de todo o projeto.

A secção 4.1 descreve, em termos gerais, a planificação e métodos usados para desenvolver o código. A secção 4.2 detalha, com pormenor, o funcionamento de cada programa, com o auxílio de fluxogramas e exposição das diversas mensagens trocadas entre os vários programas.

4.1 Planificação e Metodologia

No desenvolvimento de um projeto é sempre necessário fazer uma boa planificação e utilização de ferramentas que permitam fazer uma gestão eficiente, não só das pessoas alocadas ao projeto, mas também dos recursos e do tempo.

Nesta secção é apresentado o processo seguido na gestão, documentação, desenvolvimento e implementação do código elaborado no âmbito desta dissertação.

4.1.1 *Agile Software Development*

A metodologia *Agile* é uma alternativa aos métodos tradicionais de gestão de projetos, tipicamente usados em desenvolvimento de *software*. Esta metodologia ajuda as equipas a responderem a imprevistos através de incrementos de cadência iterativa, designados *sprints*.

Nesta dissertação foi usada a metodologia *Scrum*, que é uma forma simplificada da metodologia *Agile*, por ser simples e flexível. Dado que, na presente dissertação, apenas existe um participante no desenvolvimento do código necessário à implementação do sistema IAP, foi necessário ajustar o *Scrum* à medida das necessidades particulares do desenvolvimento individual. Como ferramenta de auxílio à organização das tarefas a realizar, foi utilizada uma plataforma *online* designada *Trello*.

4.1.2 Documentação do Código

Por forma a manter um *workflow* suave e contínuo, torna-se necessária a elaboração de documentação e sua constante atualização. De notar que esta documentação pode assumir, posteriormente, um papel importante, quer na manutenção do código, quer na eliminação de um *bug*, por exemplo. Existem vários programas que permitem documentar no próprio código, através do uso de comentários, que depois podem ser exportados para uma página *html* ou para *pdf*.

O código foi escrito na linguagem de programação C e o programa utilizado na compilação da sua documentação designa-se por Doxygen.

4.1.3 *Coding Style*

Quando existe mais do que um programador, envolvido num projeto, é importante que haja harmonização no estilo de escrita do código usado, de modo a tornar mais fácil a sua leitura e compreensão. No entanto, esta preocupação deve estar presente mesmo existindo apenas um programador, pois, para além de ser importante uma boa organização do código, podem sempre existir contribuições futuras de terceiros.

Para garantir que todo o código está formatado de forma consistente, foi utilizado um programa designado por *indent*, que tem como objetivo varrer os ficheiros de *input* e colocá-los com a formatação previamente especificada num ficheiro de configuração. O *coding style* adotado inspira-se num misto do *GNU style* e do *Kernighan & Ritchie style*.

4.1.4 Ferramentas de Auxílio ao Desenvolvimento do IAP

No desenvolvimento de um projeto composto por vários ficheiros de código e dividido por pastas, torna-se difícil compilar todos os ficheiros um a um, e fazer o *link* dos mesmos. Para facilitar esta tarefa foram criadas *Makefiles*, que são ficheiros com uma sintaxe específica, que comanda a compilação de todos os ficheiros e o *link* entre eles. Através do uso da ferramenta *Make*, é possível compilar *software* a partir do seu código fonte, independentemente do seu tamanho, do número de ficheiros envolvidos ou da organização dos diretórios.

Importa referir, por um lado, que foram criadas *Makefiles* de forma hierarquizada, sendo que a *Makefile* principal de cada programa tem como função chamar todas as outras existentes em cada diretório, e, por outro, que foram criados vários *targets*, de forma a poder compilar o código, instalá-lo, desinstalá-lo ou criar um ficheiro comprimido com todo o código, pronto a ser distribuído.

4.1.5 Controlo de Versões

O uso de controlo de versões é justificável até quando existe apenas um programador, mas quando o número de programadores é igual ou superior a dois, torna-se absolutamente crítico controlar rigorosamente o trabalho efetuado por cada um deles, por forma a evitar a perda de código escrito.

O uso do controlo de versões permite garantir a integridade de todo o trabalho realizado, funcionando como uma espécie de *backup*. Quando se faz o *deployment* do código

para produção, esta transição pode ser tão simples quanto executar apenas um comando e todas as novas funcionalidades ficam *online* imediatamente.

Para fazer o controlo de versões do código, foi utilizado o programa *subversion*, sendo que o repositório usado foi o da Universidade de Aveiro (code.ua.pt).

4.1.6 *Logging* e Constante Monitorização

O uso de *logs* e a constante monitorização dos mesmos permite identificar, mais facilmente, problemas que ocorram na aplicação, agilizando respostas para correções de *bugs*. A monitorização dos *logs* permite ter uma atitude proativa, antecipando eventuais problemas com que os utilizadores possam vir a ser confrontados.

Os *logs* permitem, ainda, identificar o que correu mal quando a aplicação apresenta algum problema. Assim, se um utilizador reportar um *bug*, indicando a data e a hora, o programador pode rapidamente detetar a sua origem, desde que a falha tenha sido registada nos *logs* do sistema.

Todos os programas desenvolvidos, no âmbito desta dissertação, contêm mensagens de erro detalhadas e divididas por categoria, que são enviadas para o servidor *Syslog*.

4.1.7 Segurança

A segurança dos sistemas de informação é um aspeto de elevada importância na atualidade. Cada vez mais, os negócios estão dependentes da *internet* para se desenvolverem e as empresas são obrigadas a confiar os seus dados confidenciais. Apesar de haver empresas inteiramente especializadas em segurança, que fazem testes de penetração de modo a identificar pontos com falhas, todas as aplicações que têm exposição na *internet* devem ser desenvolvidas de raíz de forma eficaz para travar possíveis ataques.

Todo o código desenvolvido nesta dissertação teve especial atenção a *bugs* que pudessem originar acesso não autorizado como, por exemplo, através de *buffer's overflows*.

Para além disso, o acesso à página *web*, para gestão do sistema IAP, não é permitido a partir do exterior. Para que o utilizador possa aceder à página de gestão do sistema IAP, necessita primeiro de se ligar à rede por VPN. Nesta dissertação, foi utilizado o *OpenVPN* para configuração de uma VPN, mas qualquer outro tipo de VPN pode ser usado.

4.1.8 Medição de desempenho e Otimização

A medição de desempenho do código é fulcral no desenvolvimento de programas que tenham uma grande interação ou que corram em sistemas limitados.

Para fazer o *profiling* dos programas desenvolvidos, foram usadas as ferramentas *Valgrind* e *gprof*, que permitem fazer *debugging* à memória e analisar o uso do processador por função, podendo avaliar se existem *memory leaks* ou se uma determinada função do código está a demorar demasiado tempo a executar uma tarefa que deveria ser rápida.

4.1.9 Documentação do Uso dos Programas

A documentação do uso de um programa é importante na medida em que o seu utilizador precisa de saber como utilizar o programa.

Para todos os programas desenvolvidos nesta dissertação foram escritas *man pages*, nas quais se detalha o uso de todos os parâmetros utilizados pelos programas, bem como o formato dos ficheiros de configuração usados pelos mesmos.

De modo a que o programa *man* consiga interpretar corretamente as *man pages*, é necessário que estas usem os chamados *troff formatting commands*.

O *troff* é um sistema de *typesetting*, criado em 1970, pelos mentores do sistema operativo *Unix*. O ficheiro *troff* pode ser escrito manualmente ou gerado, usando uma multiplicidade de ferramentas, como o *DocBook* ou *PerlPOD*.

As *man pages* dos programas desta dissertação foram todas escritas manualmente.

4.1.10 Configuração de Parâmetros dos Programas

Existem duas formas de passar parâmetros para um programa, ou através da linha de comandos, ou a partir de um ficheiro de configuração.

No desenvolvimento dos programas desta dissertação, optou-se pela configuração dos parâmetros através de um ficheiro de configuração, permitindo, assim, alterar os parâmetros e guardar as modificações num ficheiro, de forma a que, cada vez que o programa inicializa, possa ler os vários parâmetros do ficheiro e comportar-se de acordo com os mesmos. Esta é uma prática corrente no desenvolvimento de programas, especialmente quando estes são *daemons* e, como tal, não têm qualquer interação com o utilizador.

Estes ficheiros de configuração permitem que o instalador configure os diversos parâmetros de modo a que o sistema IAP funcione em qualquer tipo de rede, não estando dependente de uma determinada arquitetura. Assim, resulta um *design* geral que pode ser ajustado de acordo com a rede onde estiver instalado.

4.1.11 Início Automático dos Programas

Todos os programas desenvolvidos nesta dissertação não necessitam de interação direta com o utilizador e não têm *standard input* ou *output* (*daemons*). Para que os programas sejam inicializados logo que o sistema arranque, foram criados *scripts init* e *upstart*.

O *upstart* é um substituto do *daemon init* do *Linux*, tendo sido entretanto criado um novo sistema denominado de *systemd*, no qual é necessário conceber um outro tipo de ficheiro, designado *unit*. Este ficheiro segue a lógica dos ficheiros do *upstart*, especificando qual é o programa que se pretende correr, bem como em que circunstâncias e que tipo de monitorização se pretende ter. Por outro lado, os *init scripts* são *scripts* escritos em *bash*, em que o programador tem que escrever todo o código para validar um conjunto de parâmetros e criar condições específicas, entre outros.

De referir que existe, normalmente, em todas as distribuições *Linux* um ficheiro *template*, chamado *skeleton*, que contém grande parte do código que é comum a todos os *init scripts*.

4.2 Funcionamento dos Programas Desenvolvidos

Os programas desenvolvidos, no âmbito desta dissertação, foram os seguintes:

- Para a infraestrutura de rede tecnológica:
  - *Discovery Suite*: *Discovery Notifier* (DN) e *Discovery Center* (DC)
  - *IntAPcom* (IAPc);
  - *IntAPcomConv* (IAPcC); e,
  - *Web Page*.
- Para operar serviços:
  - *Digital I/O* (Dio): *Shutter commander* (Sc).

No desenvolvimento do sistema *Integrated Access Point* surgiu a necessidade de manter uma lista permanentemente atualizada, com todos os IAPs conectados à rede. Esta necessidade determinou o desenvolvimento de um conjunto de programas, designados *discovery suite*, que têm como único objectivo fazer o *tracking* de todos os IAPs na rede.

A atribuição de um IP a um dispositivo ligado à rede pode ser feita de forma estática (*Static* IP), em que o utilizador configura os parâmetros manualmente, ou de forma dinâmica (*Dynamic* IP), em que os parâmetros são configurados de forma automática através de troca de mensagens entre o dispositivo e um servidor DHCP, conforme a seguir se descreve:

- *Static* IP: O uso de IPs estáticos em redes locais é uma solução muito usada quando se pretende que um servidor tenha um IP bem definido e que nunca se altere. Por exemplo, numa LAN, quando se utiliza a tecnologia NAT, é possível ao *router* identificar o servidor ao qual se destina o pacote de inicialização de comunicação, uma vez que não há alteração do IP no servidor. Inicialmente, a *internet* e a rede *Ethernet* não tinham outra forma de alocar um endereço IP a uma máquina que não fosse através do uso estático de endereços. O utilizador configurava a máquina para usar um determinado IP, e mais ninguém, na mesma rede privada, podia configurar a sua máquina para usar o mesmo IP. Esta forma de alocar endereços IP tornava difícil a gestão da rede, especialmente se esta tivesse uma grande dimensão como, por exemplo, uma rede instalada num *Campus* Universitário ou numa grande empresa. Qualquer pessoa que tentasse conectar-se à rede com o seu computador pessoal teria que ter conhecimentos básicos sobre os protocolos usados em redes *Ethernet* e seria ainda necessário solicitar ao administrador da rede permissão para usar um determinado IP. No caso do *Integrated Access Point*, o uso de *static IP* seria uma opção, mas pouco robusta, uma vez que qualquer alteração das configurações no *router* poderia automaticamente desconectar todos os APs da rede, deixando a habitação sem os serviços a funcionar.
- *Dynamic* IP: O uso de IPs dinâmicos é possível através da utilização da tecnologia DHCP, que requer a existência de um servidor, que é normalmente instalado no próprio

router da rede, e um cliente que corre em todas as máquinas, tais como computadores, *smartphones* e *tablets*. Quando o utilizador tenta conectar-se a uma rede com o seu computador, depois de ter efetuado a autenticação, executa o *dhcp client*, que tem como função enviar um pacote para a rede pedindo um endereço IP. O *dhcp server* recebe este pacote e responde ao *dhcp client* com o IP a utilizar bem como uma *leasing time*, entre outras informações. A *leasing time* é o tempo de utilização que o *dhcp server* concede ao *dhcp client* para usar aquele IP. De acordo com o protocolo, decorridos 50% do *leasing time*, o *dhcp client* é obrigado a interrogar o *dhcp server* sobre se pode continuar a usar aquele IP quando o seu tempo de alocação expirar. O *dhcp server*, em princípio, responde que pode continuar a usar o IP, mas, se por algum motivo não for dada permissão, o *dhcp client*, no fim do *leasing time*, teria que pedir um novo IP. Daqui resulta a necessidade de manter uma lista permanentemente atualizada de IAPs em funcionamento numa rede, pois os endereços de IPs podem alterar-se. Neste sentido, é necessário criar um mecanismo que permita a monitorização, a todo o momento, dos IAPs instalados na rede, identificando os ativos e os que "crasharam", bem como quais os seus endereços IP.

A *discovery suite* mantém informações atualizadas dos IAPs, tais como número de IAPs instalados, o seu estado e os endereços IP, sendo constituída por dois programas, o *discovery notifier* e o *discovery center*, que são apresentados de seguida.

4.2.1 *Discovery Notifier*

O *discovery notifier* encontra-se instalado em cada um dos APs e tem como função enviar um pacote para a rede em modo *broadcast*, em intervalos de tempo definidos pelo instalador. O pacote enviado contém informações relevantes sobre o AP de onde foi expedido, tais como o endereço IP, o endereço MAC e o *hostname*. O facto de o pacote ser enviado em modo *broadcast* permite uma maior flexibilidade na configuração do programa, pois não é necessário saber, à partida, o endereço de IP do *Central Node*.

O pacote enviado tem a seguinte estrutura:

```
discovery-notifier|192.168.1.1|AA:BB:CC:DD:EE:FF|Sala
```

A informação no pacote é separada pelo carater "|" e é constituída por quatro parâmetros. O primeiro valor é estático, igual em todos os pacotes de todos os IAPs, e tem como objetivo informar o *discovery center* de que este pacote foi enviado a partir do *discovery notifier*. O segundo valor corresponde ao endereço IP, o terceiro indica o endereço MAC e o quarto corresponde ao *hostname* do IAP.

O programa é acompanhado de um ficheiro de configuração com o seguinte formato:

```
port       = 7331;
ip         = "192.168.1.255";
interval   = 3;
interface  = "eth0";
```

As quatro variáveis passíveis de serem configuradas são a porta para a qual a mensagem é enviada, isto é, a porta que o *discovery center* usa para receber as mensagens emanadas do *discovery notifier*, o endereço IP usado para enviar os pacotes, o intervalo entre cada pacote enviado e a interface (NIC – *Network Card Interface*) usada no IAP para enviar os pacotes.

De notar que, o endereço IP deve ser o endereço de *broadcast* da rede. No entanto, se, por questões de *design* da rede, existirem IAPs em redes diferentes, é possível colocar o endereço IP do *Central Node*.

O fluxograma apresentado na figura 4.1 mostra o funcionamento deste programa.

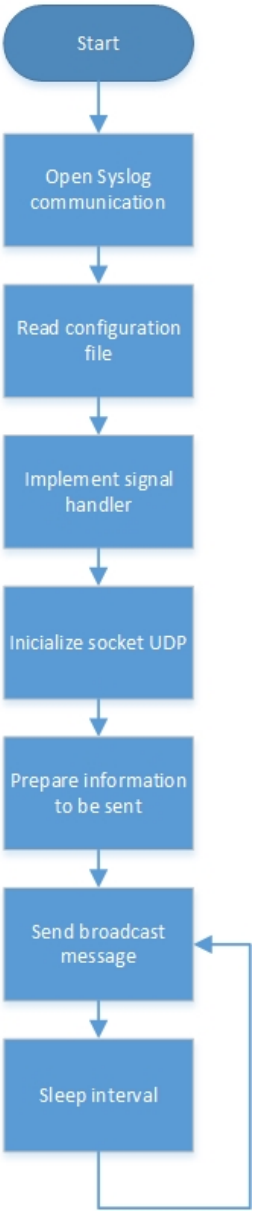


Figura 4.1: Fluxograma do *Discovery Notifier*.

O *discovery notifier* começa por efetuar a comunicação com o servidor *Syslog*, que regista os *logs* do programa e lê o ficheiro de configuração, de modo a obter os vários parâmetros configurados pelo instalador. De seguida, implementa os *signal handlers*, *SIGTERM* e *SIGHUP*, que permitem parar o programa e fazer o *reload* dos parâmetros de configuração, respetivamente. Depois, inicializa o *socket* UDP, que é usado para enviar as mensagens em *broadcast*. Finalmente, prepara a mensagem, composta pela informação descrita, e entra num ciclo infinito, que é composto por duas funções, a de enviar a mensagem em *broadcast* e a de esperar o intervalo de tempo pré-configurado no ficheiro de configuração.

4.2.2 *Discovery Center*

O *discovery center* é instalado no *Central Node* e faz a gestão da informação que recebe de todos os *discovery notifiers* instalados nos diversos IAPs. Este programa tem como objetivo recolher os pacotes enviados pelos *discovery notifiers* e compilar uma lista atualizada com a informação recebida através desses pacotes.

O programa é acompanhado por um ficheiro de configuração com o seguinte formato:

```
port = 7331;
port-con = 7332;
maxinterval = 10;
```

Este ficheiro contém informação sobre a porta usada para receber as mensagens enviadas pelo *discovery notifier*, a porta usada para receber *requests* enviados pelo *intAPcomConv* e o tempo máximo permitido a um AP sem enviar mensagens, antes de ser considerado como estando *offline*. De notar que, o intervalo máximo deve ser configurado para pelo menos três vezes o intervalo de envio de mensagens do *discovery notifier*.

O *discovery center*, para além de manter uma lista atualizada dos IAPs a correr na rede, tem ainda a função de responder aos *requests* do *intAPcomConv*. Para isso, o *discovery center* tem uma porta aberta, a partir da qual escuta as comunicações feitas pelo *intAPcomConv*. Após a conexão, o *intAPcomConv* envia um *request* ao *discovery center*, pedindo a lista dos IAPs na rede. O *discovery center* envia também a informação inerente aos IAPs que, por algum motivo, deixaram de comunicar, permitindo a visualização de que um determinado IAP deixou de responder. De notar que, a troca de mensagens entre o *intAPcomConv* e o *discovery center* é feita em JSON.

A mensagem enviada pelo *intAPcomConv* é a seguinte:

```
{
  "request": "get_ap_list"
}
```

A mensagem é composta por apenas um parâmetro, designado *request*, que contém o tipo de pedido feito ao *discovery center*. O único tipo de *request* existente é o da

mensagem apresentada, que devolve a lista de todos os IAPs existentes na base de dados do *discovery center*.

A mensagem enviada pelo *discovery center* em resposta ao *request* do *intAPcomConv* é a seguinte:

```
{
  "error": 0,
  "response": {
    "aps_list" : [
      {
        "timestatus": "green",
        "mac": "AA:BB:CC:DD:EE:FF",
        "ip": "192.168.1.36",
        "hosname": "Sala",
      }
    ]
  }
}
```

A mensagem é composta por dois parâmetros, o primeiro com o nome de *error* e com o valor 0, serve para indicar que esta mensagem não é uma mensagem de erro, e o segundo com o nome de *response*, composto por um *array* com o nome de *aps_list*, que contém tantos elementos quantos IAPs existirem na base de dados do *discovery center*. Neste exemplo, por questões de simplificação, apenas se mostra a informação de um IAP.

O *array* é composto por objetos que contêm quatro parâmetros de informação relevante a cada IAP. O primeiro parâmetro com o nome de *timestatus* pode assumir três estados: *green*, *yellow* ou *red*. Se for *green* significa que o IAP está a funcionar e a comunicar, se for *yellow* quer dizer que o IAP falhou o envio da mensagem pelo menos uma vez, mas ainda se encontra dentro do tempo máximo configurado pelo *discovery center* para designar o IAP como estando *offline*, e se for *red* mostra que o IAP não respondeu dentro do tempo máximo configurado e, por isso, é considerado como estando *offline*. Os outros três parâmetros contêm o endereço MAC, o endereço IP e o *hostname* do IAP.

Caso ocorra algum erro na construção da informação a enviar para o *intAPcomConv*, o *discovery center* envia uma mensagem de erro com o seguinte formato:

```
{
  "error": 1,
  "error_message": "Error on receiving request"
}
```

A mensagem é composta por dois parâmetros, o primeiro com o nome de *error*, que é constituído por um número inteiro que identifica o tipo de erro, e o segundo com o nome de *error_message*, que contém uma mensagem breve que explica o erro, sendo de referir que existem outros tipos de erros e respetivas mensagens.

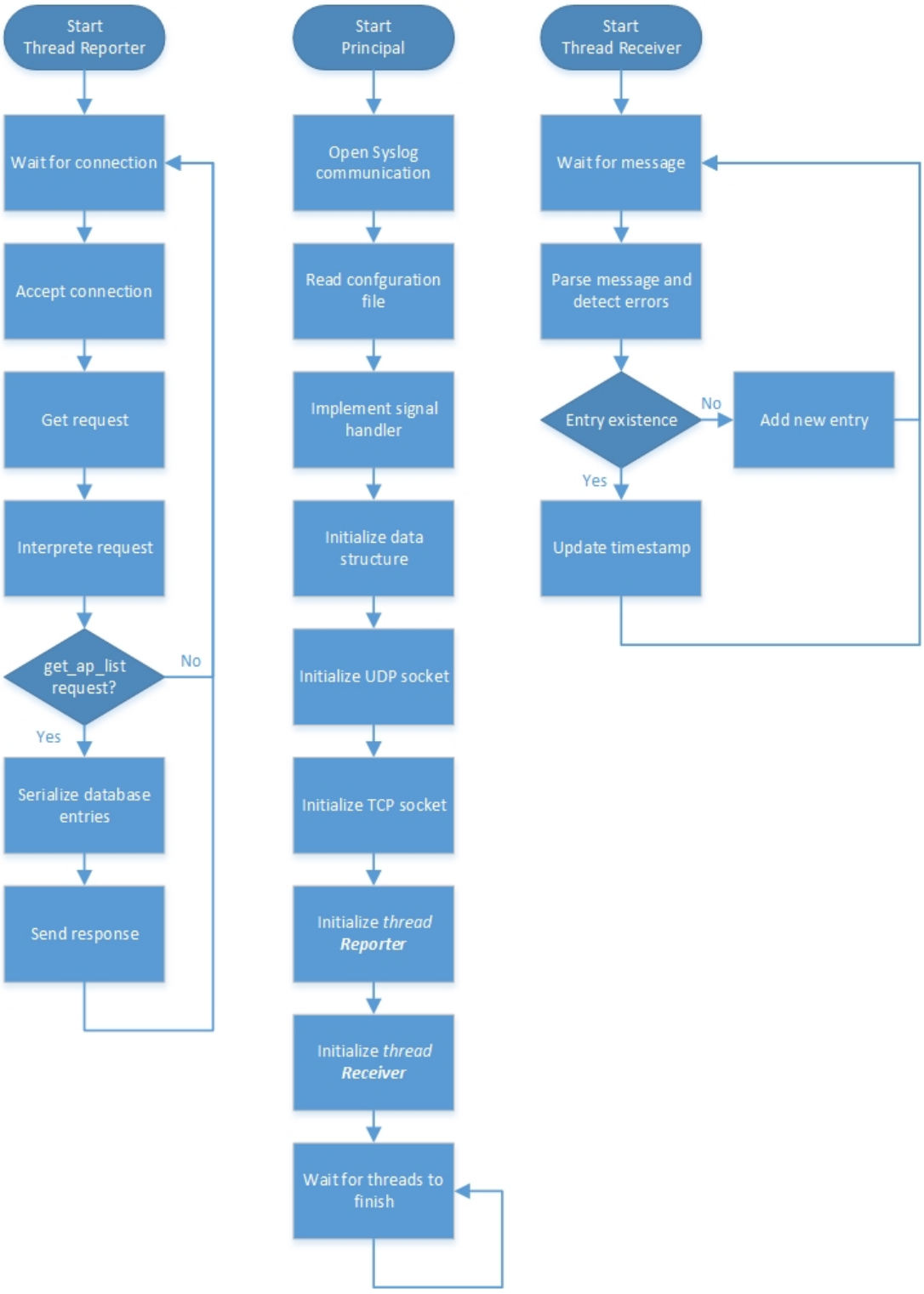


Figura 4.2: Fluxograma do *Discovery Center*.

A figura 4.2 apresenta o fluxograma que mostra o funcionamento deste programa.

À semelhança do *discovery notifier*, o programa *discover center* inicializa a comunicação com o servidor *Syslog*, que regista os *logs* do programa, lê o ficheiro de configuração para obter os vários parâmetros configurados pelo instalador e implementa os *signal handlers*, *SIGTERM* e *SIGHUP*, que permitem parar o programa e fazer o *reload* dos parâmetros de configuração, respetivamente. De seguida, inicializa a estrutura de dados que é composta por uma lista ligada, bem como o *socket* UDP, que recebe as notificações do *discovery notifier*, e o *socket* TCP, que recebe os *requests* enviados pelo *intAPComConv*. Finalmente, inicializa as *thread reporter* e *receiver*, que têm como função satisfazer os pedidos do *intAPcomConv* e receber os pacotes enviados pelo *discovery notifier*, respetivamente.

A *thread reporter* espera que o *intAPcomConv* se conecte e, de seguida, aceita a conexão, lê o *request* efetuado e interpreta o pedido. O único tipo de *request* implementado é o *get_ap_list* e, por isso, é o único tipo de pedido que pode ser feito. Os dados que estão na estrutura de dados são serializados em JSON e enviados para o *intAPcomConv*.

A *thread receiver* recebe as mensagens emanadas do *discovery notifier* e interpreta a informação contida dentro da mensagem. Caso a mensagem seja emanada de um IAP que já se encontre na estrutura de dados, apenas é atualizado o *timestamp*, caso este não se encontre na estrutura de dados, é adicionada uma nova entrada com a informação do IAP.

4.2.3 *IntAPcom* (IAPc)

Resolvida a questão de manter uma base de dados com todos os IAPs ligados à rede, é necessário resolver o problema decorrente do facto de cada serviço precisar de uma porta para comunicar com os eventuais clientes. Por forma a minimizar o número de ligações TCP, foi desenvolvido o programa *intAPcom*, que funciona como um comutador de mensagens dentro de cada IAP. Este programa tem a função de receber mensagens enviadas pelo *intAPcomConv* e entregá-las ao serviço correspondente dentro do IAP.

Assim, com o *intAPcom*, o cliente liga-se a uma só porta e, através de comunicação com recurso ao JSON, informa a ordem de serviço que pretende ver executada e a quem se destina. De seguida, o *intAPcom* faz o *forwarding* dessa mesma ordem para o serviço correspondente. Acresce que o *intAPcom* foi desenvolvido de forma modular, de maneira a suportar, em diferentes etapas, os novos serviços que venham a ser adicionados, bem como novas formas de comunicação, sejam elas por *Bluetooth*, *ZigBee* e *Irda*, entre outros.

O *intAPcom* é acompanhado de um ficheiro de configuração de suporte à implementação do sistema IAP, que permite ao instalador mudar vários parâmetros da instalação, com o seguinte formato:

```
port = 8331;
port-unix = "/tmp/intapcomunix"
```

As variáveis que podem ser alteradas são a porta que fica à escuta de comunicações com origem no *intAPcomConv* e a porta lógica UNIX à qual todos os serviços que correm num determinado IAP se vão ligar.

O funcionamento deste programa é explicitado nos fluxogramas representados nas figuras 4.3 e 4.4.

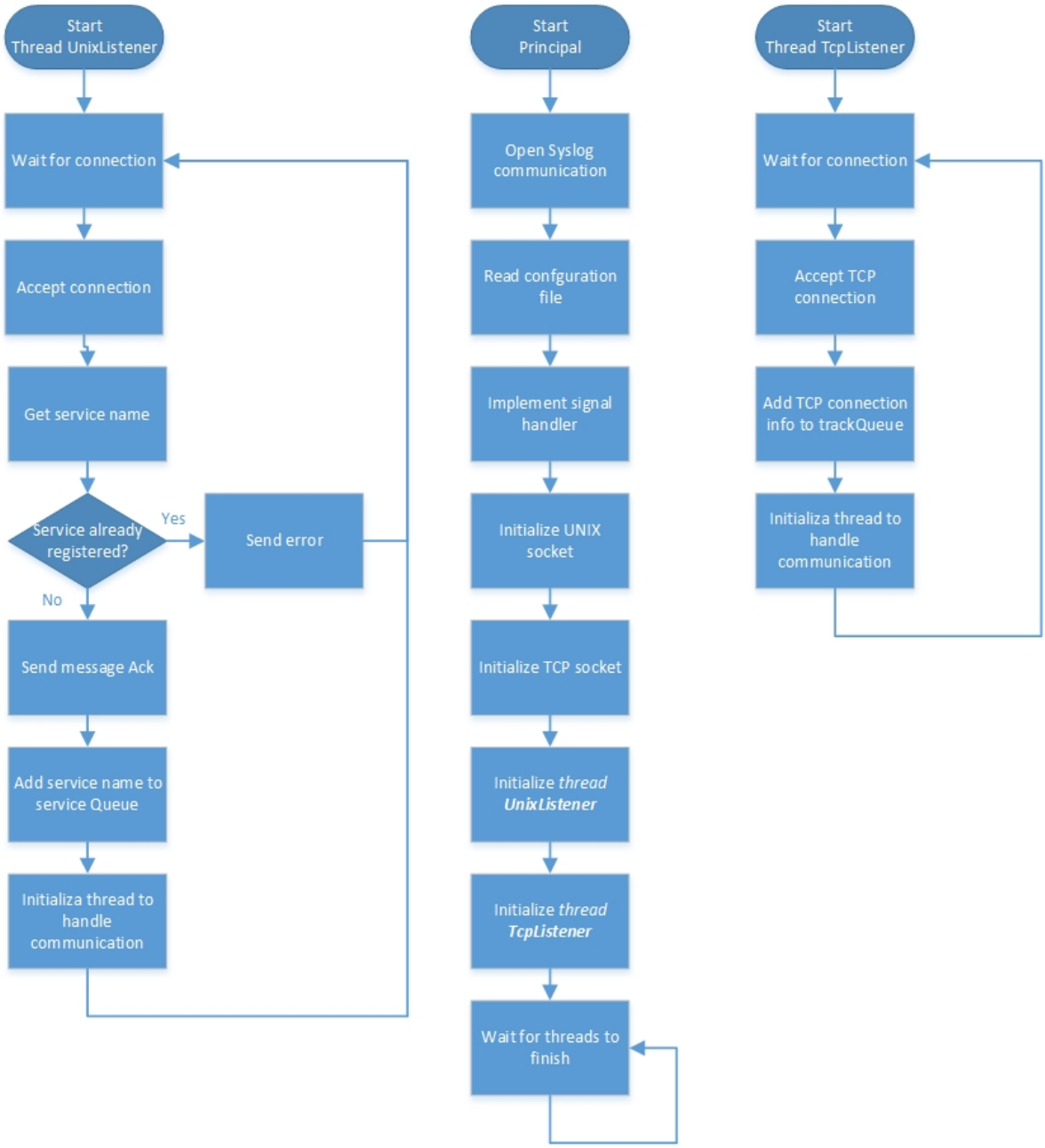


Figura 4.3: Fluxograma do *IntAPcom* (1).

Quando um serviço começa a correr, procura conectar-se ao *intAPcom* numa porta cujo ID está definido no ficheiro de configuração. O *intAPcom*, quando inicializa, começa duas *threads*: uma corresponde ao *socket* TCP, para comunicações com origem num cliente na rede, e a outra corresponde ao *socket* UNIX, para comunicações com programas que estejam a correr única e exclusivamente na mesma máquina.

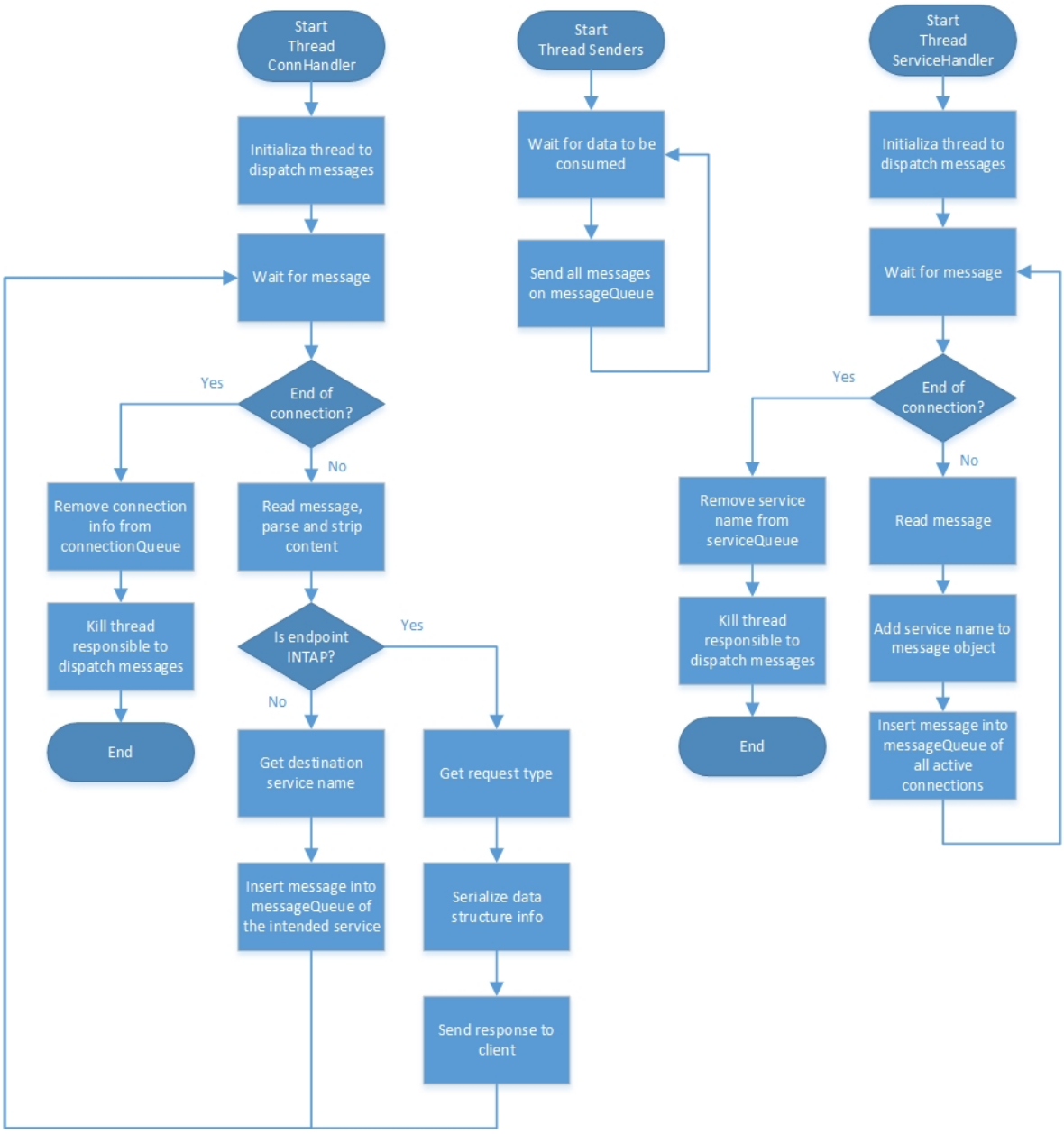


Figura 4.4: Fluxograma do *IntAPcom* (2).

Ambas as *threads* são responsáveis por inicializar, correr e monitorizar uma porta de comunicação. A *thread* responsável por inicializar, correr e monitorizar a *socket* TCP, é aquela que recebe todos as ligações efetuadas por clientes na rede, sendo que:

- Apesar de só estar implementada a comunicação via TCP/IP, com poucas alterações seria possível acrescentar novas formas de comunicação com o IAP como, por exemplo, através de *Bluetooth*, *ZigBee* ou *IrDA*;
- Quando um cliente se liga ao IAP, é-lhe atribuida uma nova porta, permitindo assim que, na eventualidade de haver mais do que um cliente a querer ligar-se ao mesmo

tempo, o *intAPcom* consiga fazer a gestão e permita a ligação de novos clientes;

- Mesmo não havendo nenhuma identificação por parte do cliente, o *intAPcom* regista numa base de dados a sua comunicação com o mesmo, e desta forma, independentemente do número de clientes ligados, o *intAPcom* tem sempre o registo na base de dados com o número, tipo e estado dos clientes ligados; e,
- Quando um cliente se desconecta, o *intAPcom* remove a entrada correspondente a esse cliente e toda a informação a ele associada.

A *thread* responsável por monitorizar a *socket* UNIX fica à espera que algum serviço se conecte a ela, sendo que, a partir desse momento, atribui uma nova porta ao serviço para que possa continuar a comunicação. De seguida, ocorre um *handshake* entre o *intAPcom* e o serviço, no qual este último se identifica fornecendo informação relevante. O *intAPcom* regista essa informação numa base de dados que contém todos os serviços que estão a correr no IAP e que efetuaram uma ligação com o *intAPcom*, podendo dessa forma encaminhar, corretamente, para cada serviço as mensagens provenientes da rede. De notar que, quando uma mensagem chega ao *intAPcom*, essa mensagem pode ter como destino qualquer um dos serviços, ou até mesmo o próprio *intAPcom*.

As mensagens que chegam ao *intAPcom* apresentam o seguinte formato:

```
{
  "endpoint": "INTAP",
  "request": "list"
}
```

De notar que, neste exemplo, o *endpoint* é o próprio *intAPcom*, mas pode ser qualquer outro serviço que esteja a correr no IAP. No caso concreto, foi efetuado um *request* para obter a lista de todos os serviços a correrem no IAP, mas também se podia ter efetuado um *request* para saber o número de serviços a correr ou perguntar, especificamente, se um determinado serviço está a ser executado no IAP.

Um exemplo de resposta dada pelo *intAPcom*, é o seguinte:

```
{
  "intapcom":{
    "endpoint": "INTAP",
    "response": [
      "shuttercommander",
      "dummyservice",
      "audioservice"
    ]
  }
}
```

Na resposta é identificada a origem da mensagem, que neste caso é o *intAPcom*, e o parâmetro "*response*" contém um *array* com o nome de todos os serviços a correr no IAP em questão.

Para manter um fluxo de informação estável e garantir que a transmissão das mensagens é bem sucedida, o *intAPcom* foi implementado com o uso de um mecanismo de *queueing*, que permite a cada mensagem que chega, ser inserida na *queue* correspondente ao serviço a que se destina, retirando o *overhead* com a informação do destino. Para além disso, este mecanismo permite, mais tarde, implementar *Quality of Service*, se assim se pretender. Uma vez que se utilizam *queues*, é possível, facilmente, atribuir prioridades às mensagens, de forma a obter um controlo mais rigoroso sobre que mensagens devem ter prioridade em eventuais congestionamentos.

Neste sentido, o serviço recebe apenas a ordem para executar, sendo que, se o serviço se desconectar do *intAPcom*, significa que "crashou". De referir, ainda, que existe um programa que supervisiona os serviços, que ao detetar que terminaram, volta a inicializá-los. Assim, é necessário que o *intAPcom* seja capaz de lidar com o facto de um serviço se poder desconectar e agir de modo a repor a ligação. O *intAPcom*, aquando da desconexão de um serviço, remove a informação correspondente na base de dados de serviços conectados, procedendo à limpeza de qualquer recurso usado naquela ligação. Quando o serviço se volta a conectar, é repetido todo o procedimento atrás descrito. Assim, garante-se que qualquer serviço possa ser conectado ou desconectado a qualquer momento, pois o *intAPcom* funciona apenas como um mediador, cuja responsabilidade é entregar as mensagens ao serviço em concreto. Quando algum serviço pretende notificar sobre um determinado evento, envia a mensagem para o *intAPcom* e este, por sua vez, reencaminha a mensagem a todos os clientes a ele conectados naquele instante.

Com a utilização do programa *intAPcom* simplifica-se o desenvolvimento dos serviços, pois não existe mais a preocupação com o modo como as comunicações ocorrem na rede, podendo assim ter uma noção abstrata da interface do serviço com o resto da rede, enviando simplesmente notificações e tratando as mensagens recebidas como ordens.

De notar que, para qualquer mensagem recebida, apenas chega a ordem e nunca o remetente da mesma, pois o serviço não necessita de saber quem a emitiu, mas simplesmente qual é o tipo de ordem. Quando o serviço envia uma notificação, o *intAPcom* agrega à mensagem, qual a origem de tal notificação, pois o cliente pretende saber quem foi que o notificou de um determinado evento. Por exemplo, se alguém passar na frente de um sensor de movimento e esse sensor enviar a notificação para o *intAPcom* que está a correr no IAP, o *intAPcom* reencaminha a notificação juntamente com a identificação pertinente para que o cliente possa saber qual dos sensores foi acionado.

Como o *intAPcom* faz o roteamento local das mensagens, a única informação da origem que remete é o nome de serviço. Na secção seguinte, é demonstrado que o *intAPcomConv* agrega, ainda, o IP associado ao IAP que corre o *intAPcom* que o notificou. Desta forma, o cliente fica com a informação completa sobre a origem da mensagem, qual dos IAPs (através do endereço IP) e, entre eles, qual dos serviços reportou o incidente (através do nome do serviço que vai agregado à mensagem).

Para facilitar a serialização ou desserialização da informação, bem como a adição ou remoção de informação a uma determinada mensagem, foi usada a tecnologia JSON que facilita a gestão e organização das mensagens que circulam não só na rede, mas também no *intAPcom* que se encontra nos diversos IAPs.

4.2.4 *IntAPcomConv* (IAPcC)

Depois de ultrapassada a questão de manter uma lista com os IAPs na rede e o problema do uso excessivo de recursos num determinado IAP, é necessário conceber um controlador (*Central Node*), que se ligue a todos os IAPs da rede e faça uma gestão cuidada dos dados e mensagens, de modo a que seja possível apresentar, numa só ligação, toda a informação que todos os IAPs disponibilizam. Neste sentido, foi desenvolvido o programa *intAPcomConv*, cuja função se baseia na gestão das ligações aos IAPs na rede, constituindo um só ponto de ligação para o cliente, pelo que este pode requerer qualquer tipo de informação de qualquer serviço.

Os parâmetros existentes no ficheiro de configuração são os seguintes:

```
port-websocket = 9331;
port-intapcom = 8331;
port-discovery = 7332;
```

O primeiro parâmetro refere-se à porta onde os *browsers* se vão ligar por *websocket*, o segundo à porta usada pelo *intAPcomConv* para se ligar ao *intAPcom* e o terceiro à porta usada pelo *intAPcomConv* para se ligar ao *discovery center*.

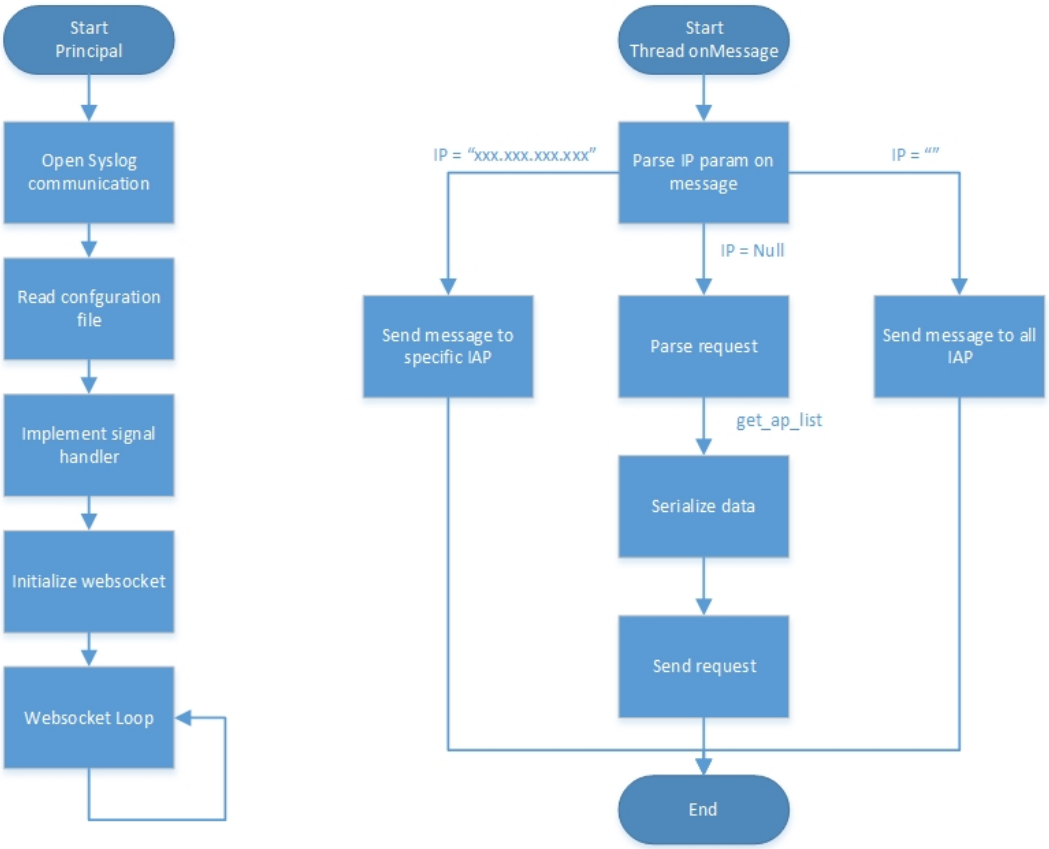


Figura 4.5: Fluxograma do *IntAPcomConv* (1).

As figuras 4.5 e 4.6 apresentam os fluxogramas de funcionamento deste programa.

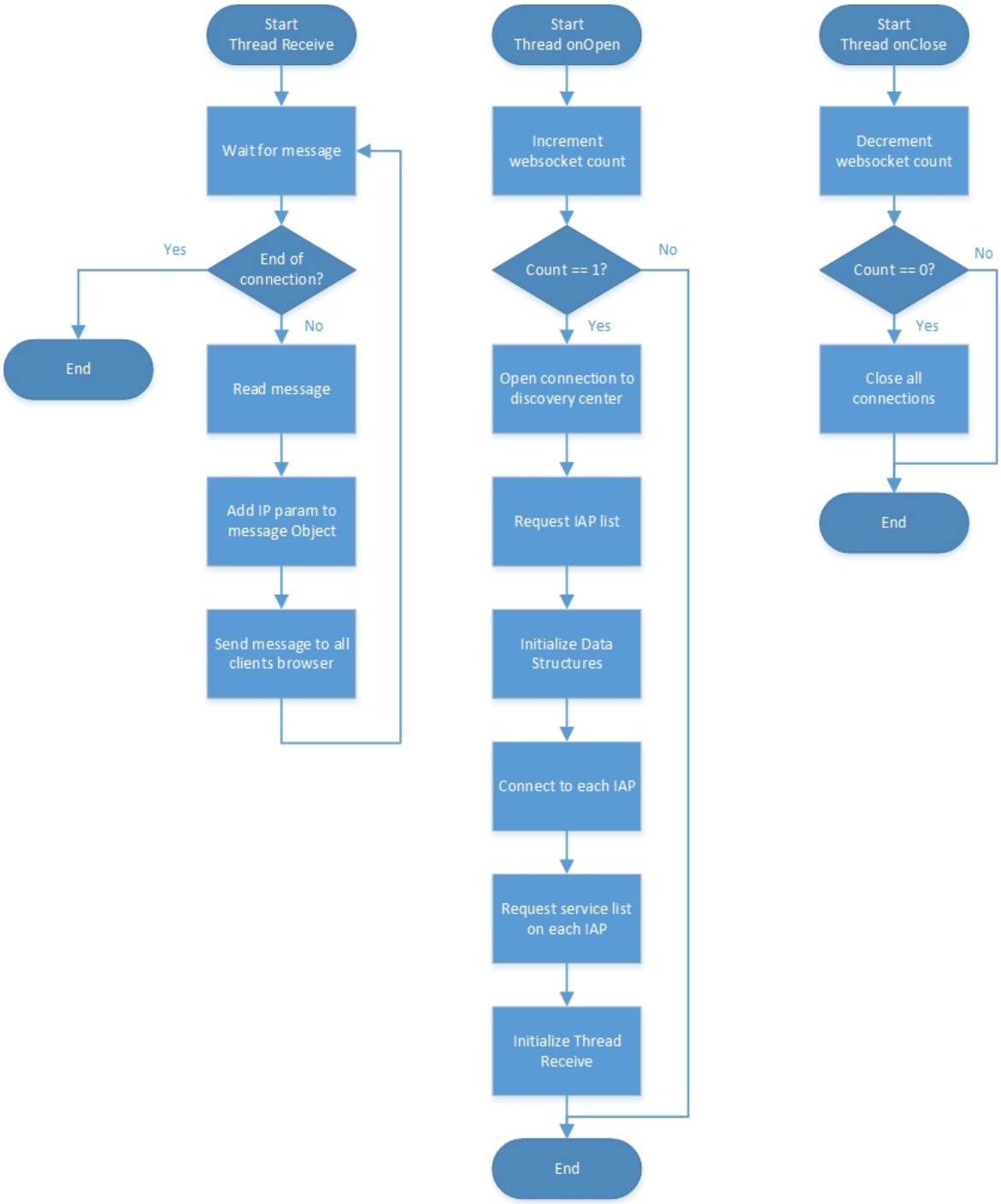


Figura 4.6: Fluxograma do *IntAPcomConv* (2).

Quando o *intAPcomConv* é inicializado, não estabelece nenhuma ligação, uma vez que as ligações aos IAPs da rede só são estabelecidas após o primeiro contacto do cliente. Se alguém se ligar ao *intAPcomConv* e se esta ligação não tiver sido precedida de qualquer

outra, então o *intAPcomConv* liga-se ao *discovery center* e faz o pedido da lista dos IAPs na rede. Após obter esta lista, o *intAPcomConv* liga-se diretamente a cada um dos acessos fazendo o *request* dos serviços ativos em cada um deles. Terminada esta rotina de inicialização, o *intAPcomConv* passa a ter uma base de dados com as informações relevantes de todos os IAPs ativos na rede e dos serviços que neles correm.

A partir deste momento, qualquer mensagem que seja proveniente do cliente ou de qualquer um dos IAPs, é transmitida ao devido destinatário. Sempre que um cliente pretende enviar uma mensagem a um serviço, que corre num determinado IAP, é necessário agregar à mensagem, o nome do serviço e o IP do IAP onde corre esse mesmo serviço, para que o *intAPcomConv* possa entregar a mensagem ao destinatário.

O *intAPcomConv* está desenhado para suportar a comunicação com vários clientes, em simultâneo, e quando o último cliente se desconecta, o *intAPcomConv* desconecta-se de todos os IAPs, eliminando as entradas da lista sobre cada um deles.

O cliente pode ser qualquer tipo de programa, desde uma aplicação para *Android/iOS*, um programa que corra no *Windows*, em *Linux* ou no *Mac OS*, uma página *web* ou até uma aplicação para a televisão. A escolha recaiu sobre a página *web*, pois, de todas as soluções, esta é a mais versátil e a que chega a um maior número de equipamentos, sendo possível acedê-la a partir de qualquer computador, a correr qualquer sistema operativo, *smartphone*, *tablet* ou televisão, isto é, desde que contenha um *browser*, qualquer dispositivo pode ser usado para comandar qualquer serviço.

4.2.5 *Web Server/Web Page*

A página *web* foi desenvolvida tendo em vista a sua modularidade e a facilidade de manuseamento. Como qualquer pessoa pode usar a rede, o acesso é efetuado através de um *login* inicial, após o qual é visualizada uma página *web* com um menu do lado esquerdo e o conteúdo do lado direito.

No menu, o primeiro item mostra a lista dos IAPs na rede e os restantes correspondem aos serviços a correr nos IAPs, sendo que cada item corresponde a um serviço.

Para efeitos de comunicação entre o *browser* e o *intAPcomConv*, foi usada uma tecnologia denominada *websockets*, que permite ao *browser* interagir com um servidor através de *sockets*. Esta tecnologia torna possível a notificação de qualquer acontecimento, em tempo real, por exemplo, se um sensor de passagem detetar algum movimento, envia de imediato uma notificação. Com a tecnologia *websocket*, o *browser* não tem que fazer um *request* ao servidor para obter nova informação, pois esta é, imediatamente, transmitida ao *browser* sem nenhum *request*.

Em alternativa a esta tecnologia, poder-se-ia utilizar um método chamado *polling*, que consiste no *browser* fazer *requests* sucessivos ao servidor, com um determinado intervalo de tempo, até obter a informação. Mas, nestas circunstâncias, se o intervalo fosse demasiado longo, o evento poderia acontecer nesse espaço temporal e, quando o *browser* fizesse o *request*, a informação recebida poderia já não ser atual e, por outro lado, se o intervalo fosse demasiado curto, a página *web* iria exigir excessivos recursos ao *browser* que, por sua vez, iria utilizar o CPU de forma intensiva.

De seguida, apresenta-se um exemplo de *request* feito pelo *browser* ao serviço *Shutter Commander*:

```
{
  "ip": "192.168.1.1"
  "data": {
    "service": "shuttercommander",
    "request": "SET BAR 50%"
  }
}
```

Esta mensagem é composta por dois parâmetros, o primeiro corresponde ao IP do IAP para o qual se pretende enviar a mensagem e o segundo corresponde aos dados que serão interpretados pelo *intAPcom*. Este último parâmetro contém dois parâmetros adicionais, a identificação do serviço para o qual se destina a mensagem e a mensagem que se pretende enviar para o serviço. Neste caso, a mensagem é enviada para o serviço *Shutter Commander*, que se encontra no IAP com o IP 192.168.1.1, e a mensagem informa o serviço de que se pretende colocar o estore a 50% da posição absoluta.

4.2.6 *Digital I/O* (Dio): *Shutter Commander* (Sc)

Para demonstrar o correto funcionamento da infraestrutura, foi desenvolvido um serviço do tipo *Digital I/O*, que se baseia no comando de um estore elétrico, podendo o utilizador, através da página *web*, comandar o movimento do estore para subir ou descer. O serviço consiste num programa que corre como *daemon* e que começa por se conectar à porta em que o *intAPcom* escuta. Depois de estabelecida a comunicação, o serviço inicia um *handshake* com o *intAPcom* podendo, assim, enviar ou receber mensagens.

O serviço recebe mensagens do tipo:

```
1) UP
2) DOWN
3) STOP
4) SET BAR 10%
```

As três primeiras mensagens são ordens diretas para subir, descer ou parar o estore, respetivamente. A quarta mensagem permite um controlo mais preciso, através da especificação da posição final do estore.

O fluxograma deste programa é apresentado na figura 4.7.

Dado que este serviço se baseia no controlo de um estore, é também necessário um componente de *hardware* que se destina ao controlo do motor. Este componente apenas faz a conversão das tensões e permite a passagem de uma maior corrente para que o motor consiga rodar.

Por fim, importa referir que para desenvolver serviços adicionais, basta escrever um programa que comunique com o *intAPcom*, criar a página *web* correspondente ao serviço pretendido e adicioná-la ao *web server*.

No capítulo seguinte demonstra-se o funcionamento do sistema *Integrated Access Point* e comprova-se a eficácia da infraestrutura da rede tecnológica desenvolvida.

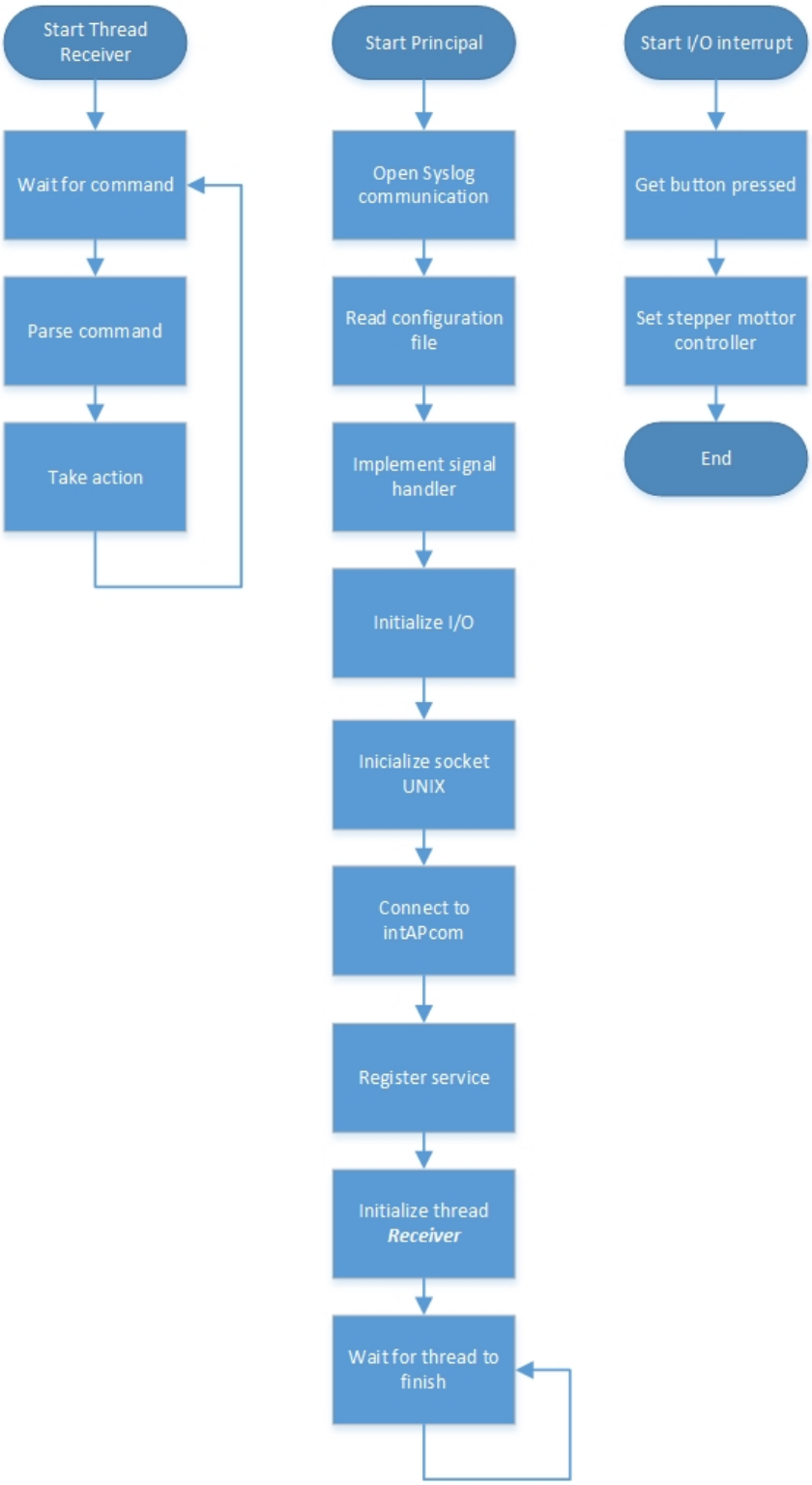


Figura 4.7: Fluxograma do *Shutter Commander*.

Capítulo 5

Demonstração do Sistema

Neste capítulo pretende-se demonstrar o funcionamento do sistema IAP, definindo-se para o efeito uma rede com a estrutura do esquema da figura 5.1.

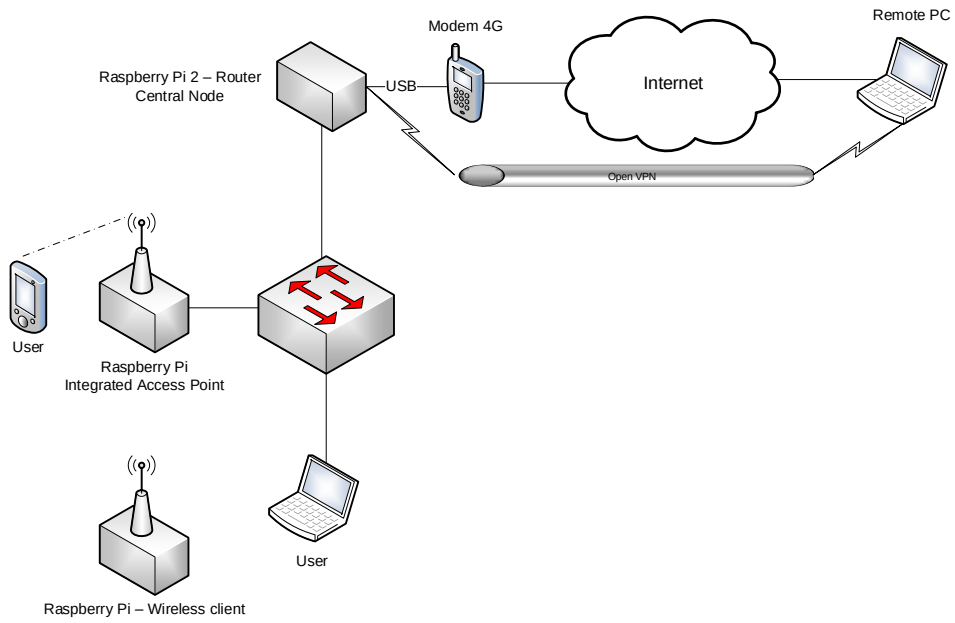


Figura 5.1: Esquema da Rede.

As principais componentes físicas da infraestrutura da rede tecnológica são o núcleo do IAP (*Central Node*) e as interfaces aos dispositivos (*access points*). O *Raspberry Pi* (ver figura 5.2), assume uma elevada importância neste sistema, por ser um elemento comum ao *Central Node* e ao *access point*.

A rede é composta por um *Raspberry Pi* (RPi2) que desempenha as funções de um *router*, um *switch unmanaged*, um *Raspberry Pi* (RPi1) que funciona como *access point*, um outro *Raspberry Pi* (RPi3) que funciona como cliente *wireless* e um computador.

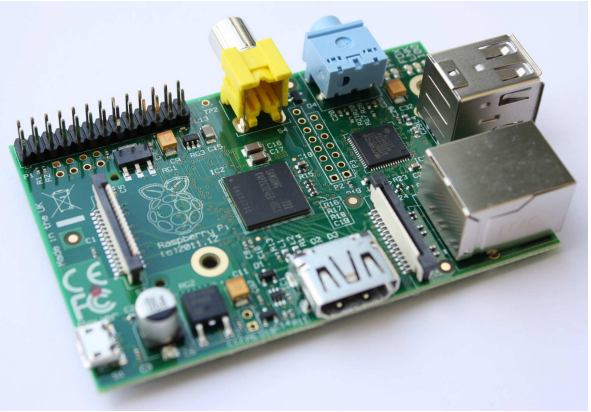


Figura 5.2: *Raspberry Pi* (extraído de [28]).

Dado que o RPi2 funciona como *router*, a sua ligação à *internet*, através da porta WAN, pode ser feita de várias formas. O RPi2 pode ligar-se diretamente a um *modem* ADSL, a um ONT, ou até a um *smartphone* que disponibilize *internet*, através da partilha por USB.

Para efeitos de demonstração, o RPi2 está ligado a um *smartphone Android*, através de um cabo USB. Como o *smartphone Android* atribui IPs por DHCP, o RPi2 recebe um IP privado, a partir do qual se testa a ligação à *internet*. De notar que o *smartphone* faz NAT no roteamento dos pacotes, o que deve ser tido em conta quando se habilitar o servidor *OpenVPN*. Uma vez que o *smartphone* já faz NAT, disponibilizando assim uma interface para uma rede local, seria expectável que em vez de se usar o RPi2 como um *router*, se usasse como um *switch*. No entanto, o objetivo passa por demonstrar como seria a estrutura de uma rede em condições normais, isto é, em que o equipamento ligado à porta WAN do RPi2 é um *Network Demarcation Device* (NDD) que não faz NAT e que usa IPs públicos para comunicação.

O RPi2, para além de fazer NAT na sua porta WAN, tem também um servidor DHCP que atribui IPs a todos os dispositivos na rede. Adicionalmente, corre um servidor DNS, que tem como função fazer *relay* de todos os pedidos DNS para um *DNS server* exterior como, por exemplo, o da *Google*. No RPi2 encontra-se instalado o servidor *OpenVPN*. Se o RPi2 fosse ligado a um NDD e tivesse um IP público na sua porta WAN, não seria necessário fazer mais nada, mas, neste caso, como o que está ligado à porta WAN do RPi2 é um *smartphone* que faz NAT, é necessário abrir a porta UDP 1194 e apontar para o IP privado da porta WAN do RPi2. Assim, quando um cliente, que se encontre noutra rede, tentar conectar-se com o servidor *OpenVPN*, o *smartphone* redireciona todos os pacotes recebidos na porta 1194 para o IP do RPi2.

Como o RPi1 está configurado para funcionar como um *switch* entre a porta LAN e a porta WLAN, com a ajuda do programa *hostapd*, que corre em *userland*, é possível tornar o RPi1 num *access point*. Com esta configuração os pacotes são encaminhados devidamente entre os dispositivos que estiverem ligados por *wireless* ao RPi1 e os dispositivos ligados à rede por cabo. O RPi3 está configurado para funcionar como um cliente *wireless*, que se liga ao RPi1 e disponibiliza os seus serviços via *wireless*. Como tal, é

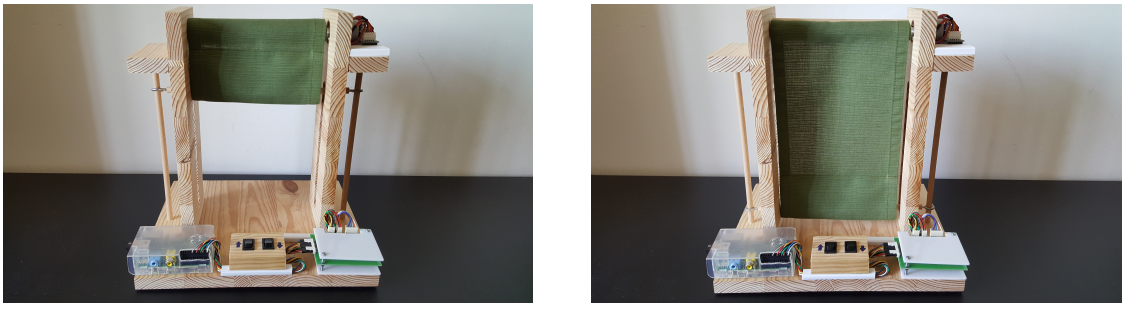
possível disponibilizar serviços onde não existem cabos de rede.

O computador ligado na rede tem apenas a função de garantir que existe conectividade entre os vários dispositivos e gerir os serviços domóticos instalados no RPi1 para controlar o estore. Existe, ainda, outro computador ligado noutra rede, que tem instalado o *OpenVPN* e que o usa como cliente, para se poder conectar ao servidor *OpenVPN*, de forma a criar um túnel VPN para aceder à página de gestão dos serviços domóticos.

Depois de criada a rede e configurados os dispositivos para exercerem as suas funções, é necessário instalar os diversos programas criados no âmbito desta dissertação, de modo a utilizar a rede com a implementação de serviços de domótica. Assim, o RPi2, que funciona como *router* neste esquema, tem a função de *Central Node* na infraestrutura do IAP. De notar que não é necessário que o equipamento que tem a função de *Central Node* seja o *router*. O RPi1 funciona como *access point* e, por isso, tem todos os serviços de domótica que controlam os aparelhos de uma habitação. Para efeitos de demonstração, apenas se ligou um *Raspberry Pi* que funciona como *access point*, mas poderiam ser ligados tantos quantos necessários.

Neste projeto desenvolveu-se apenas um programa de suporte à operação de serviços dentro da habitação, para testar o funcionamento da infraestrutura da rede, mas, no domínio da domótica, existe uma vasta variedade de programas que operam serviços. O serviço implementado designa-se por *Shutter Commander* e tem como objetivo comandar um estore e mostrar o seu estado. Para isso, instalou-se o serviço no RPi1 para comunicar com o estore e a respetiva *tab* da página *web* que controla este serviço.

Na figura 5.3 apresentam-se as imagens do protótipo de um estore, desenvolvido no âmbito da presente dissertação, através do qual foi possível provar o funcionamento do sistema IAP.



(a) (b)

Figura 5.3: Protótipo de um Estore.

A construção do protótipo, em madeira, foi executada sob desenho *Autocad*, com o apoio de uma empresa fabricante de mobiliário, utilizando para o efeito uma máquina CNC (controlo numérico), que permitiu o rigor de cortes e rasgos na madeira, facilitando o funcionamento do estore sem quaisquer interferências. A motorização, o controlo e a sinalização do estore, que inclui todos os dispositivos e componentes acopladas, foram instalados pelo mestrando.

A motorização do estore é feita por um *stepper motor*, que controla com precisão o seu estado. O *stepper motor driver* foi utilizado para fornecer potência suficiente ao

stepper motor, estando ligado aos *pins I/O* do RPi1. O programa que opera este serviço interage com os *pins* de forma a fazer subir ou descer o estore. Como o *stepper motor* permite controlar os passos dados em cada rotação, é possível manter com precisão o nível a que o estore se encontra. Este programa interage com o *intAPCom* com o propósito de receber as ordens emanadas pelo utilizador, bem como informar em que posição se encontra o estore num determinado momento. O protótipo também contém dois interrutores que fazem subir e descer o estore. Assim, o utilizador tanto pode comandar o estore a partir destes interrutores como a partir da página *web*.

5.1 Configuração do Sistema

Nesta dissertação foram usados *Raspberry Pi* como dispositivos que permitem fazer de *access points*. No *Raspberry Pi* encontra-se instalado o sistema operativo *Linux*. Para que este funcione como um *access point*, é necessário instalar um conjunto de programas e respetivas bibliotecas e configurar os mesmos de forma a que o *Raspberry Pi* funcione como um *access point*.

Para que seja possível o acesso seguro à plataforma de gestão da rede, a partir do exterior, é necessário criar um túnel VPN entre o dispositivo de onde se pretende aceder e a rede. Para isso, o *Central Node*, que é o dispositivo responsável por gerir toda a infraestrutura, é configurado como sendo um *OpenVPN server*. De notar que não é necessário que o servidor seja instalado neste dispositivo, no entanto, uma vez que este já existe na rede, e a maioria das habitações não tem servidores ligados à rede, este recurso pode ser aproveitado.

5.1.1 Configuração do *Linux* como *Access Point*

Atentendendo à versatilidade do sistema *Linux*, este foi escolhido como ferramenta para correr os programas desenvolvidos, sendo que é necessário configurá-lo para agir como um *access point*, necessitando para o efeito de duas *Network Interface Card (NICs)*, mais concretamente uma com conetor RJ45, para ligar à rede por cabo, e outra *wireless*, para servir de ponto de acesso a qualquer dispositivo que se queira ligar a esta. O sistema operativo *Linux*, depois de configurado, faz o roteamento dos pacotes entre as duas *NIC's*.

Como o IAP se baseia num *access point*, é essencial que o *Raspberry Pi* consiga funcionar como tal. Para este efeito, são necessárias duas condições, ou seja, o RPi tem que emular um *access point* com capacidade de autenticação WPA2 e tem que criar uma *bridge* entre a placa *wireless* que vai emular o *access point* e a porta *Ethernet* onde vai estar ligado à rede. Existe ainda outra forma de configuração, que consiste em criar um *NAT*, entre a interface *WLAN* e *LAN*, caso em que os clientes que se ligarem ao *access point* estão numa *subnet* diferente.

Para obter uma *bridge* em *Linux*, começa-se por criar uma interface lógica, que vai representar a *bridge*, à qual se adicionam as placas de *Ethernet* e *wireless* do *Raspberry Pi*. De seguida, remove-se os IPs atribuídos às interfaces, uma vez que essas portas vão funcionar em *Layer* 2. Por fim, inicializa-se a *bridge* e atribui-se um IP a esta interface

lógica (*bridge*), por forma a que seja possível aceder ao *Raspberry Pi* através desse IP.

Abaixo, apresenta-se um exemplo dos comandos necessários para criar uma *bridge* e adicionar as respetivas interfaces:

```
brctl addbr br0

brctl addif br0 eth0
iw dev wlan0 set 4addr on
brctl addif br0 wlan0

ifconfig eth0 0.0.0.0
ifconfig wlan0 0.0.0.0

ifconfig br0 up
ifconfig br0 192.168.100.5 netmask 255.255.255.0 up
```

Depois de criada a *bridge*, utiliza-se um *software* que vai emular um *access point*. De notar que a placa de rede *wireless* tem que suportar o modo AP. O programa que vai emular um *access point* tem o nome de *hostapd* e corre na *userland*. O *hostapd* é um *IEEE 802.11 AP*, que permite autenticar utilizadores através de *IEEE 802.1X/WPA/WPA2/EAP* e conectar-se a outros servidores autenticadores como, por exemplo, *RADIUS*.

Este programa usa um ficheiro de configuração, onde é possível especificar um conjunto alargado de parâmetros. Um exemplo de criação de um *access point* é o seguinte:

```
interface=wlan0
ssid=IAP
driver=nl80211
hw_mode=g
channel=6
ieee80211n=1
wmm_enabled=0

macaddr_acl=0
ignore_broadcast_ssid=0

auth_algs=1
wpa=3
wpa_passphrase=qwerty10
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Neste exemplo, é configurada uma rede com o nome "*IAP*" e com a *password* "*qwerty10*". A interface usada é a *wlan0*, tal como tinha sido já configurado na *bridge*, e o método de autenticação usado é o WPA2.

5.1.2 Configuração do *Linux* como Servidor *OpenVPN*

Quando o utilizador se encontra dentro da rede, este tem acesso à gestão de todos os dispositivos que nela se encontram. No entanto, quando o utilizador não se encontra dentro da rede, pode surgir a necessidade de gerir algum dispositivo remotamente, por exemplo, verificar que todas as luzes estão apagadas, enviar uma ordem de fecho de todos os estores ou ativar o aquecimento antes de chegar à habitação.

Não é de todo aconselhável que seja "aberta" a porta correspondente ao serviço de gestão, que neste caso é a porta 80 (HTTP), uma vez que deixa a rede exposta a possíveis ataques do exterior, através do acesso indevido à rede. O facto de alguém não autorizado conseguir entrar na rede, tem um impacto que não se resume apenas a conseguir o controlo sobre a habitação, já que também pode obter acesso a quaisquer dispositivos dentro da rede, tais como computadores ou servidores. Assim, é necessário utilizar um mecanismo que permita ao utilizador ter a comodidade de gerir a sua habitação quando está fora, mas também garantir a sua tranquilidade, proporcionando-lhe um sistema seguro.

A segurança pode ser garantida através das *Virtual Private Networks* (VPN), usadas há vários anos por instituições, empresas e universidades, para permitir a partilha de recursos com segurança, por utilizadores quando não estão dentro da rede. As VPNs operam, geralmente, num modelo cliente-servidor e têm como função criar interfaces virtuais e, eventualmente, *subnets* de rede virtuais, de forma a que um utilizador fora da rede possa aceder à mesma, como se estivesse dentro dela. São exemplo deste tipo de solução, o *IPSec*, o *PPTP*, o *OpenVPN* e o *L2TP*, entre outros.

Na presente dissertação, optou-se pelo *OpenVPN* por ser *free* e *open-source*. De referir que se trata de um programa de configuração relativamente fácil, sendo ainda possível criar um ficheiro de configuração para entregar aos utilizadores, de forma que estes apenas tenham que o importar para terem o túnel configurado nos seus dispositivos.

O método de encriptação usado numa VPN pode ser dividido em dois grupos:

- *Symmetric Key Encryption*: este tipo de encriptação requer apenas uma chave simétrica, que é usada nos dois lados do túnel para encriptar e desencriptar as mensagens; e,
- *Public Key Encryption*: neste tipo de encriptação, cada lado do túnel tem uma chave pública e outra privada, sendo que o emissor encripta a mensagem com a chave pública do recetor e o recetor desencripta a mensagem com a sua própria chave privada. Como apenas o recetor tem a sua chave privada, apenas este pode desencriptar a mensagem.

O *OpenVPN* suporta os dois métodos de encriptação. Para esta demonstração, foi usado o método de chave simétrica, uma vez que é mais fácil de configurar e não requer nenhum *Certificate Authority*. No entanto, com este método apenas é possível servir um cliente de cada vez. De notar que o *OpenVPN* usa a porta UDP 1194 para efetuar as conexões entre o cliente e o servidor, por isso, é necessário abrir esta porta na *firewall* ou no *router* que esteja a aplicar NAT na interface WAN.

Para configurar o *OpenVPN*, é possível criar-se um ficheiro de configuração com todos os parâmetros necessários à criação do túnel VPN e depois iniciar o programa, apontando

para o ficheiro de configuração, ou passar todos os parâmetros de configuração pela linha de comandos. Neste caso, o *OpenVPN* vai ser configurado pela linha de comandos.

Utilizando o método de encriptação simétrico, o primeiro passo consiste em gerar uma chave simétrica, através do seguinte comando:

```
openvpn --genkey --secret chave.key
```

O ficheiro "chave.key" será gerado com a chave simétrica. Este ficheiro tem que existir tanto no dispositivo do cliente como no servidor, pois é uma chave partilhada. Na figura 5.4 apresentam-se os IPs utilizados para as ligações do túnel, bem como os IPs internos.

Figura 5.4: *Exemplo da Implementação da Virtual Private Network.*

De seguida, inicializa-se o *OpenVPN* do lado do servidor com o seguinte comando:

```
openvpn --dev tun --ifconfig 192.168.0.1 192.168.0.2 \
        --secret chave.key
```

O parâmetro "--dev tun" informa o *OpenVPN* que se pretende usar uma interface virtual TUN, o parâmetro "--ifconfig" especifica os endereços IPs usados em cada um dos lados do túnel e o parâmetro "--secret" passa o ficheiro "chave.key" ao *OpenVPN* para ser usado como chave de encriptação. O *OpenVPN* cria uma interface TUN, com o IP 192.168.0.1, e disponibiliza a ligação de um cliente *OpenVPN* a partir do endereço 192.168.0.1.

Por fim, executa-se o cliente com o seguinte comando:

```
openvpn --remote 10.0.0.1 --dev tun \
        --ifconfig 192.168.0.2 192.168.0.1 --secret chave.key
```

O parâmetro adicional, "--remote 10.0.0.1" informa o *OpenVPN* para se ligar àquele IP, na porta 1194. Desta forma, consegue-se estabelecer um túnel VPN entre o cliente e a rede onde o sistema IAP está instalado.

5.2 Comercialização do Sistema *Integrated Access Point*

No sentido de avaliar a viabilidade económica deste projeto, definiu-se a instalação simbólica para dez serviços a serem disponibilizados numa habitação com catorze divisões e um pátio exterior com jardins, conforme apresentado no apêndice C. Deste modo, temos um *access point* em cada divisão da habitação e mais um no exterior, instalado na garagem para controlar os três jardins, portões de garagem, portas de acesso ao pátio da habitação, lavandaria e arrumos. O *Central Node*, instalado na divisão do escritório, funciona também como interface aos dispositivos aí existentes.

Para demonstrar a modularidade do sistema IAP foram estabelecidas quatro hipóteses, consoante as divisões da habitação e os respetivos serviços definidos para cada divisão, tendo-se verificado que o custo total da instalação pode variar entre 360 e 2.350 euros, sendo o seu custo médio de 1.125 euros. Por fim, calculou-se o custo para implementação do sistema IAP, com apenas o serviço de estores em duas divisões (escritório e sala) da habitação, dado que no âmbito desta dissertação apenas foi desenvolvido este serviço, e verificou-se que, com apenas 360 euros, pode ser instalado um serviço de controlo de estores. A análise apresentada no apêndice C demonstra que o sistema desenvolvido, denominado IAP, é substancialmente mais barato que os sistemas de domótica que utilizam os protocolos *standard* (*KNX* e *LON*).

Capítulo 6

Conclusões e Trabalho Futuro

Neste capítulo apresentam-se as principais conclusões e alguns tópicos de desenvolvimento futuro.

6.1 Conclusões

Apesar dos benefícios proporcionados pela domótica e do aumento de popularidade entre os consumidores, existem ainda alguns problemas associados a esta tecnologia que podem constituir oportunidades de mercado para quem for capaz de apresentar soluções alternativas, importando destacar o elevado custo de alguns serviços disponibilizados e a dificuldade de implementação de soluções de domótica em habitações já construídas, que limitam o acesso a grande parte dos potenciais utilizadores.

A arquitetura de rede desenvolvida neste projeto, designada por *Integrated Access Point*, é composta por um conjunto de programas da infraestrutura da rede (*discovery suite*, que inclui *discovery notifier* e *discovery center*, *intAPcom*, *intAPcomConv* e *web page*) que servem de suporte ao programa que opera o serviço de comando de estores (*Digital I/O*: *Shutter Commander*).

A infraestrutura do *Integrated Access Point* requer que seja mantida uma base de dados atualizada, a todo o instante, com todos os *access points* conectados à rede, tendo, por isso, sido desenvolvido um pacote de ferramentas, designado por *discovery suite*, cujo propósito consiste em fazer o *tracking* de todos os *access points* na rede. Adicionalmente, o facto de cada serviço precisar de uma "linha de dados", ou seja, uma linha lógica pela qual os dados passam e que são as ordens enviadas ou as notificações recebidas pelos *access points*, resulta na criação do *intAPcom*. Após resolver o problema de manter uma lista atualizada com os *access points* conectados à rede e o problema do uso de excessivos recursos num determinado *access point*, falta um controlador que se ligue a todos os *access points* e que faça a gestão cuidada dos dados e mensagens, de modo a que seja possível apresentar numa só ligação toda a informação que todos os *access points* disponibilizam. Para este efeito, é desenvolvido o *intAPcomConv* cuja funcionalidade consiste na gestão das ligações aos *access points* conectados à rede e que se apresenta como o único ponto de ligação ao cliente. Relativamente ao conjunto de programas da infraestrutura da rede, importa, por fim, referir que o interface entre

o utilizador e o sistema é baseado numa *web page* concebida como uma ferramenta de *layout* simples e dispondo de um menu principal com a lista dos serviços disponíveis, que permite ao utilizador interagir com o sistema de forma permanente e em tempo real.

Para provar a viabilidade do sistema foi desenvolvido um programa que opera o serviço de comando de estores (*Digital I/O*: *Shutter Commander*), tendo, neste contexto, sido criado um protótipo através do qual se demonstra o correto funcionamento da infraestrutura da rede tecnológica. De notar que, para desenvolver futuros serviços, apenas é necessário escrever um programa que comunique com o *intAPcom* e criar a *web page* correspondente ao serviço, para ser adicionada ao *web server*, e a nova entrada no menu principal com a lista dos serviços disponíveis.

Os objetivos propostos para a presente dissertação foram atingidos com sucesso, pois o facto de a solução desenvolvida (*Integrated Access Point*) ser baseada na comunicação via *Ethernet* (TCP/IP) permite aproveitar a cablagem existente ou comunicar através de *wireless*, reduzindo consideravelmente os custos de implementação, sendo também de destacar, por um lado, que a infraestrutura da rede tecnológica utilizada torna possível que todos os serviços funcionem de forma correta e independentemente da estrutura da habitação e/ou do número de serviços já implementados, o que aumenta a modularidade desta solução, e, por outro, que o interface utilizado é baseado numa *web page*, através da qual o utilizador comanda todos os serviços instalados na sua habitação, aumentando a comodidade proporcionada por esta solução.

De uma forma geral, os protocolos *standard* (KNX e LON) com as suas ligações aos fabricantes de componentes elétricas, certificados por normas internacionais, estão na origem de comportamentos no mercado que dificultam a divulgação da tecnologia e, por isso, constituem um entrave à expansão da domótica, ficando a sua divulgação circunscrita a nichos de mercado com elevado poder de compra. O exemplo comparativo, apresentado no apêndice C, mostra, com base na recolha de preços de mercado, que os sistemas de domótica que utilizam protocolos *standard* (KNX e LON) são cinco a seis vezes mais caros que o sistema desenvolvido com base no *Integrated Access Point*.

6.2 Trabalho Futuro

Durante a elaboração da presente dissertação foram identificados alguns aspetos que podem ser desenvolvidos em trabalhos futuros, sendo de destacar:

- A alteração do funcionamento da *discovery suite*, de modo a que esta se possa auto configurar na rede sem necessidade de configurações prévias. No projeto desenvolvido, a configuração da *discovery suite* tem de ser realizada por um técnico, porém é possível que o *discovery center* envie periodicamente uma trama com a informação pertinente, ou seja, o IP da máquina em que está a correr e a porta em que escuta, entre outros, para que o *discovery notifier* se possa configurar com base nesta informação.

- A criação de novos mecanismos de comunicação com o *intAPcom*, tais como o *Bluetooth*, o *Irda* e o *Zigbee*, entre outros. Neste projeto, o *intAPcom* recebe informação por TCP/IP, sendo esta ligação o único ponto de comunicação, pelo que ao adicionar

novos tipos de comunicação se potencia a criação de uma infraestrutura maior. Por exemplo, se o *intAPcom* tiver um ponto de comunicação *Bluetooth*, é possível criar um aplicativo para telemóvel (iOS, Android ou Symbian, entre outros) que comunica com o *intAPcom* por *Bluetooth* e mostra a informação ao utilizador de forma organizada e apelativa.

- A junção do *Central Node* e do *router* num só dispositivo, pois apesar de, no projeto desenvolvido, o *Central Node* se apresentar como um dispositivo separado, também é possível que este seja embutido no *router*, de modo a agregar todas as funcionalidades apenas num dispositivo. Todas as redes necessitam de um *router*, que contém um *web server*, um *DHCP Server* e um *DNS Server*, entre outras aplicações, pelo que seria possível integrar no *router* os serviços que correm no *Central Node*, designadamente o *discovery center* e o *intAPcomConv*, sendo que as *web pages* poderiam ser providenciadas pelo mesmo servidor que o *router* já tem instalado para efeitos de configuração de rede.

- A simplificação da instalação, pois, neste projeto, todos os dispositivos têm de ser ligados aos *access points* através de um cabo, para que ocorra troca de informação, mas a comunicação pode ser efetuada por *wireless*, caso se adicione *endpoints* com *Zigbee* nos *access points* e nos dispositivos a controlar. Assim, em vez de existirem fios do *Integrated Access Point* até ao serviço implementado para comandar, por exemplo, o motor instalado no estore, é possível estabelecer esta comunicação via *Zigbee*, não sendo necessário fazer rasgos na parede.

Para além dos aspetos mencionados, que procuram simplificar a infraestrutura da rede tecnológica e o acesso dos utilizadores, importa referir que podem ainda ser desenvolvidos outros serviços, conforme abordado aquando da apresentação das conclusões.

Lista de Referências

[1] P. Matutino, *Conceção e Desenvolvimento de uma Rede Domótica*, Dissertação para a Obtenção do Grau de Mestre em Engenharia Eletrónica e de Computadores, Instituto Superior Técnico da Universidade Técnica de Lisboa, 2001.

[2] A. Chamusca, *Domótica & Segurança Eletrónica*, Ordem dos Engenheiros, 2006.

[3] Automação e Controlo, Available: http://fenixautomacao.blogspot.pt/2011/03/automacao-residencial-domotica.html, Accessed: October, 2015

[4] Fuchsia Research Corporation, *A Brief Technology Overview of the Lighting Control Marketplace*, Home Automation White Paper, 2014.

[5] Wago, *Native KNXnet/IP devices*, Scientific Conference Paper, 2006

[6] Powerline - Vantagens e Desvantagens, Available: http://www.palmtop.pt/powerline-vantagens-e-desvantagens/, Accessed: October, 2015

[7] C. Withanage, C. Yuen e K. Otto, *A comparison of the popular home automation technologies*, ResearchGate Conference Paper, 2014

[8] Hermann Merz, *Building Automation: Communication systems with EIB/KNX, LON and BACnet (Signals and Communication Technology)*, Springer, 2009.

[9] O que é o KNX, Available: http://www.seas.es/blog/automatizacion/que-es-knx/, Accessed: October, 2015

[10] Protocolo LON, Available: http://www.serconint.com/tac_schneider_electric_partner.php, Accessed: October, 2015

[11] Dave Evans, *How the Next Evolution of the Internet Is Changing Everything*, Cisco White Paper, 2011

[12] A. Becker, G Sénéclauze, P. Purswani e S. Karekar, *Internet of Things*, Atos White Paper, 2012

[13] Wendell Odom, *CCNA Routing and Switching 200-120 Official Cert Guide*, Cisco Press, 2013.

[14] J. Gouveia e A. Magalhães, *Redes de Computadores*, FCA, 2005

[15] L. B. Sousa, *TCP/IP Básico e Conectividade em Redes*, Editora Érica, 2008

[16] E.B. Kelly, *Quality of Service In Internet Protocol Networks*, Prepared for the International Communications Industries Association, 2002.

[17] *Power-over-Ethernet Overview*, Transition Networks White Paper, 2015

[18] Power over Ethernet, Available: http://www.cablinginstall.com/articles/print/volume-18/issue-11/features/technology-maximizing-the-efficiency-of-high-power-power-over-ethernet.html, Accessed: October, 2015

[19] *Power over Ethernet*, Axis Communications White Paper, 2005

[20] A. Mannan, D.K. Saxena e M. Banday, "A Study on Power Line Communication"in *International Journal of Scientific and Research Publications*, Volume 4, Issue 7, July 2014, pp. 1-4.

[21] Entenda melhor o PLC, Available: http://blog.ccna.com.br/2009/09/07/entenda-melhor-o-plc-power-line-communications/, Accessed: October, 2015

[22] J. Gerhart, *Home Automation and Wiring*, McGraw-Hill Publishing Co, 1999.

[23] D. S. Bhojane, S. R. Chaudhari, P.D. More e E.G. Rajgure, "Power Line Communication"in *International Journal of Engineering Research and Applications*, Volume 2, Issue 1, Jan-Feb 2012, pp. 747-753.

[24] Anna-Greta Nystrom e Fredrik Hacklin, "Operator Value-Creation Through Technological Convergente: The Case of VoIP", in *16th European Regional Conference*, Porto, Portugal, 2005.

[25] Barrie Sosinsky, *Cloud Computing Bible*, Wiley Publishing, Inc, 2011.

[26] Cloud Computing, Available: https://www.oficinadanet.com.br/artigo/internet/cloud-computing-o-guia-basico-para-leigos, Accessed: October, 2015

[27] P.K. Tiwari, e Dr. B. Mishra, "Cloud Computing Security Issues, Challenges and Solution"in *International Journal of Emerging Technology and Advanced Engineering*, Volume 2, Issue 8, Aug 2012, pp. 306-310.

[28] Raspberry Pi, Available: http://www.tenettech.com/product/2184/ tenet-technetronicsraspberry-pi, Accessed: October, 2015

[29] R. Sá, *Sistemas e Redes de Telecomunicações*, FCA, 2007.

[30] F. Boavida, M. Bernardes, P. Vapi, *Administração de Redes Informáticas*, FCA, 2009.

Apêndice A

Topologias de Rede

Em termos gerais, é possível dizer que a topologia de rede é um mapa de rede, que pode ser de dois tipos [14]:

- Topologia física: ligada ao meio de comunicação, *layout* dos computadores e componentes da rede; e,
- Topologia lógica: relacionada com a forma como os *hosts* comunicam entre si, através dos meios disponíveis, define como é que a informação circula na rede.

As principais topologias de rede são [14] [29]:

Bus

O barramento (*bus*) é uma topologia em que todos os computadores são ligados num mesmo barramento físico de dados. Apenas uma máquina pode transmitir no barramento num dado momento e, consequentemente, todas as outras "escutam" e recolhem para si os dados que lhes são destinados. Quando um computador está a transmitir um sinal, toda a rede fica ocupada, e se outro computador tentar enviar outro sinal ao mesmo tempo, ocorre uma colisão e é preciso reiniciar a transmissão. [14]

Esta tecnologia em barramento (*bus*) utiliza um único cabo (*backbone*) coaxial que é terminado em ambas as extremidades (barramento simples), sendo todos os *hosts* diretamente conectados a este *backbone*.

As principais vantagens desta topologia são a facilidade de instalação, o facto de ser relativamente económica e utilizar menos cabo do que as outras topologias. Como desvantagens, importa referir a dificuldade de mudar ou mover nós (dispositivos ligados à rede) e de diagnosticar falhas ou erros, bem como a ausência de tolerância a falhas, pois a rede deixa de funcionar caso falhe um dos nós.

Mesh

Na topologia em malha (*mesh*) existe uma ligação direta entre cada um dos nós, ou seja, todos comunicam com todos, sendo de referir uma variante desta estrutura utilizada na *internet*, que é a malha híbrida. A principal vantagem desta estrutura está relacionada com a capacidade de cada elemento dispor de caminhos alternativos ao utilizar outros elementos como pontos de trânsito até ao destino final. De notar que esta

estrutura apresenta limitações quando o número de pontos a interligar ultrapassa algumas dezenas, podendo tornar-se impraticável, embora seja utilizada no núcleo das redes, na interligação dos nós principais, onde é necessário garantir um nível elevado de disponibilidade.

Star

A topologia em estrela (*star*) é usada para interligação de um número elevado de pontos, que se ligam a um ponto central, que é um dispositivo que pode ser um *hub* ou um *switch* e que atua como um concentrador. Em relação à estrutura anterior, importa referir a existência de elementos adicionais, designadamente os comutadores. Na prática, esta estrutura é repetida de uma forma hierárquica, de modo que existam estrelas que interliguem os pontos centrais da estrela de nível mais baixo. As principais vantagens da topologia em estrela são a facilidade de modificação do sistema, pois todos os cabos convergem para um só ponto, a simplicidade do protocolo de comunicação, que se resume a selecionar qual o nó periférico que em cada momento está ligado ao nó central, o facto de existir um dispositivo por derivação, logo, se este falhar, apenas esse dispositivo é afetado, e de ser fácil a deteção e isolamento de falhas, dado que o nó central está diretamente ligado a todos os outros. No entanto, também existem desvantagens como o maior comprimento do cabo, para efetuar as ligações, e a dependência do nó central, uma vez que, se este falhar, a rede fica inoperacional.

Ring

A topologia em anel (*ring*) pode ser considerada como uma versão simplificada das redes em malha. Nesta estrutura, cada posto está diretamente ligado a dois outros postos da rede e os dados circulam no sentido de um posto para o outro, sendo que cada posto inclui um dispositivo de receção e transmissão, o que lhe permite receber o sinal e transmiti-lo ao posto seguinte, no caso da informação não lhe ser destinada. No que se refere a vantagens, importa destacar o pequeno comprimento do cabo, a simplicidade do desenho das cablagens e a ausência de necessidade de armários de distribuição de cabos, uma vez que as ligações são efetuadas em cada um dos nós. As desvantagens desta topologia estão relacionadas com a falha de um nó provocar a falha na rede e com as dificuldades de localização de falhas e reconfiguração da rede, bem como a dificuldade no estabelecimento de protocolo de acesso à rede, dado que cada nó tem de assegurar a continuidade da informação e só depois pode enviar a sua própria informação, após certificação de que a rede está disponível.

Tree

A topologia em árvore (*tree*) apresenta uma estrutura simples, na medida em que se traduz na difusão do mesmo sinal desde um ponto central para todos os terminais, sendo que, nos pontos de divisão, o sinal é repetido para cada um dos troços até se atingirem os extremos. Inicialmente, estas estruturas foram usadas para difusão de sinais de televisão, no entanto, atualmente, foi introduzido o sentido ascendente, do utilizador para a rede, passando a ser possível o fornecimento de serviços bidirecionais, como o serviço telefónico ou o acesso à *internet*. No entanto, muito embora seja possível comunicar nos dois sentidos, a capacidade de transmissão no sentido descendente é, normalmente, maior do que no sentido ascendente.

Apêndice B

Redes de Computadores

As redes de computadores podem ser classificadas segundo vários critérios, sendo que a classificação baseada na abrangência geográfica levou à classificação das redes em três grandes categorias [30]:

- Redes de área local (LAN): interligam estações de trabalho, periféricos, terminais e outros dispositivos num único prédio ou área geograficamente limitada, permitindo o acesso a meios de grande largura de banda e conectividade ininterrupta;
- Redes de área metropolitana (MAN): interligam equipamentos abrangendo uma cidade ou conjunto de povoações próximas umas das outras; e,
- Redes de área alargada (WAN): interligam dispositivos em grandes áreas geográficas, operando ao nível de um país ou continente.

Adicionalmente, seguindo a mesma lógica, existem outras categorias como as redes de área pessoal (PAN), que abrangem o espaço vizinho de um indivíduo e incluem os seus dispositivos, ou as redes de armazenamento (SAN), que abrangem um conjunto de equipamentos e de armazenamento em massa, normalmente localizados num *datacenter*.

Os principais dispositivos para o funcionamento de uma rede são [14] [15]:

Hub

O *hub* é um dispositivo que tem a função de interligar os computadores de uma rede local. O *hub* funciona como um barramento compartilhado por todos os computadores, sendo que os dados enviados por um computador são recebidos por todos os outros. Os *hubs* utilizam o modo *half-duplex* e, por isso, é necessária a existência de um controlador de colisões, que se denomina por CSMA/CD. Este controlador previne colisões ao verificar se o barramento está a ser utilizado antes de começar a enviar dados. À medida que a rede se torna maior, o domínio de colisão também se torna maior e, por isso, maior é a probabilidade de haver colisões de dados. Para evitar este problema, é necessário segmentar a rede, ou seja, dividir o domínio de colisão por meio de uma *bridge* ou *switch*.

Switch

O *switch* é um aparelho que trabalha na camada 2 do modelo OSI. Tal como um *hub*, o *switch* liga vários segmentos de uma rede mas, enquanto que um *hub* distribui a

informação por todas as portas simultaneamente, o *switch* envia os dados apenas para o destinatário pretendido. Para além disso, cada porta do *switch* pertence a um domínio de colisão distinto, permitindo a utilização do modo *full-duplex*, aumentando a velocidade de transmissão e deixando de ser necessário um controlador de colisões. Este isolamento de tráfego entre redes designa-se por segmentação de redes.

Router

O *router* é um aparelho que trabalha na camada 3 do modelo OSI e tem como função interligar vários segmentos de redes diferentes. Os *routers* trabalham com protocolos de roteamento, que são protocolos de manutenção e administração de rotas de endereços de uma rede, permitindo manter tabelas de roteamento atualizadas em cada *router* para que este possa tomar as suas decisões quanto aos destinatários dos pacotes recebidos.

Firewall

A *firewall* é uma barreira de proteção entre a rede ou o computador e a *internet*, que controla o tráfego de dados, permitindo apenas o tráfego de dados autorizados. A *firewall* utiliza-se para impedir que uma rede ou um computador sejam acedidos sem autorização, sendo uma aliada no combate à pirataria informática.

Apêndice C

Estimativa de Custos do Sistema *Integrated Access Point*

A implementação e comercialização do sistema IAP, para ser mais eficiente e de fácil divulgação junto das famílias, pode utilizar serviços na *cloud*, já oferecidos por empresas, tais como servidor, *cloud* e hospedagem de *sites* em *cloud*, que funcionam como um repositório do *software* desenvolvido para cada serviço a ser comercializado.

C.1 Instalação de um Projeto Simbólico

Nesta secção aborda-se a implementação e comercialização de um projeto simbólico, quantificando custos, discutindo alternativas e abordando alguns aspetos essenciais à comercialização do sistema. Para provar a viabilidade económica definiu-se um projeto de instalação simbólico (ver tabela C.2) para uma habitação com catorze divisões e pátio exterior com jardins e para dez serviços a serem disponibilizados.

A utilização do *Raspberry Pi* é comum ao *Central Node* e aos APs, sendo que o custo de 32 euros pode ser substancialmente reduzido na medida em que não necessita, por exemplo, de GPU, conectores CSI e DSI, entre outros. Caso se desenvolva uma *custom board*, para utilização nos APs, o seu preço reduz ainda mais. Assim, temos um AP em cada uma das catorze divisões da habitação e mais um AP no exterior, instalado na garagem, para controlar os três jardins, portões de garagem e portas de acesso de pessoas ao pátio, lavandaria e arrumos. O *Central Node*, instalado na divisão do escritório, funciona também como interface aos dispositivos aí existentes. A lista de materiais necessários consta da tabela C.1.

Implementação

Com o propósito de estimar o custo de uma instalação de domótica, idealizou-se um projeto simbólico, identificando o número de divisões na habitação e mapeando os serviços a utilizar, incluindo alguns serviços de exterior. A tabela C.2 fornece essa informação.

Para efeitos de cálculo do custo da instalação, não se consideraram custos relacionados com pré-instalação (por exemplo, motorização de estores, trabalhos de construção civil na execução de rasgos nas paredes, reboco e pintura completa da divisão da habita-

Descrição dos Componentes	Unid.	Center [Quantidade]	AP's [Quantidade]	Dispositivos / Equipamentos [Quantidade]	Total [Quantidade]
Raspberry Pi	un	1,0	14,0		**15,0**
SD Card	un	1,0	14,0		**15,0**
Power Supply	un	1,0	14,0		**15,0**
USB-Wireless	un	1,0	14,0		**15,0**
Cabo de Rede	un	1,0	14,0		**15,0**
Controlador 2 saídas	un			9,0	**9,0**
Controlador 4 saídas (ou 4 canais)	un			21,0	**21,0**
Jack (1 IN/1 OUT)	un			16,0	**16,0**
Interface I2c (por un)	un			12,0	**12,0**
Cablagem	m			150,0	**150,0**
RCA Converter	un			8,0	**8,0**

Tabela C.1: Meio Físico: *Checklist* da Instalação.

Divisão da Casa/Exterior	Piso	Número de Divisões	Número de Dispositivos a controlar por tipo de serviço										
			Estoro	Audio	vídeo	Portas	Rega	Iluminação	Aquecimento	Ventilação	Segurança / intrusão	Segurança / Fugas de gás	Total
Salão	2º andar	1	1	1	1			2	1	1	1		**8**
Quartos	1º andar	4	4	4	4			9	4	4	4		**33**
Casa de banho	1º andar	3	2	3				6	3	3	2		**19**
Sala de Jantar/estar	r/c	1	2	1	1	1		4	1	1	2		**13**
Escritório	r/c	1	1	1	1			2	1	1	1		**8**
Cozinha	r/c	1	1	1	1	1		2	1	1	2	1	**11**
Casade banho	r/c	1	1	1				2			1		**5**
Hall	r/c	1		1		1		2		1	1		**6**
Caixa de escadas	r/c - 2º andar	1		1				3	1	1			**6**
Garagem	exterior	1		1		1		1		1	1		**5**
Arrumos	exterior	1				1		1		1	1		**4**
Lavandaria	exterior	1		1		1		1		1	1	1	**6**
Portão de carros	exterior	1				1							**1**
Porta de ent. Pessoas	exterior	1				1							**1**
Jardins	exterior	3					3						**3**
	Total	**22**	**12**	**16**	**8**	**8**	**3**	**35**	**12**	**16**	**17**	**2**	**129**

Tabela C.2: Projeto de uma Instalação: Dispositivos e Serviços por Divisão.

ção), sendo que, podendo o IAP comunicar com os dispositivos por *wireless*, admitiu-se nesta dissertação que utiliza cabo *Ethernet*.

A grande inovação, em termos teóricos, do desenvolvimento do sistema IAP assentou nos programas escritos, quer para a infraestrutura da rede, quer para os serviços a implementar, admitindo-se a utilização de um *access point* por divisão da habitação. Na tabela C.3 apresenta-se o custo para um projeto simbólico com as divisões e serviços pretendidos constantes da tabela C.2.

O custo total da instalação de domótica, com catorze divisões, pátio exterior com três jardins e dez serviços, ascende a cerca de 2.350 euros, não considerando quaisquer custos de pré-instalação, conforme apresentado na tabela C.4.

Comercialização

De modo a potenciar uma rápida disseminação desta tecnologia de domótica, que apresenta um baixo custo quando comparada com as existentes no mercado, entende-se que deve ser divulgada e promovida através de grande superfícies comerciais, do

Descrição dos Componentes	Unid. Medida	SBC	KNX	LON
CENTRAL NODE e AP				
Raspberry Pi	un	32,00 €		
SD Card	un	8,00 €		
Power Supply	un	7,95 €		
USB-Wireless	un	7,95 €		
Cabo de Rede	un	2,95 €		
Dispositivos				
Controlador 1 saídas (Digital I/o)	un	8,00 €	108,12 €	
Controlador 2 saídas	un	13,50 €		
Controlador 4 saídas (ou 4 canais)	un	25,00 €	256,00 €	
Jack (1 IN/1 OUT)	un	14,00 €		
Interface I2c (por un)	un	20,00 €	310,00 €	
Cablagem	m	0,13 €		
RCA Converter	un	42,00 €		
Controlador 1 saída (p/Ventilação)	un		310,00 €	
Controlador 1 saída (p/Rega)	un		130,00 €	

Tabela C.3: Preços Unitários das Componentes.

Descrição dos Componentes	Preço Unitário	Unid.	Center		AP's		Dispositivos / Equipamentos		Total	
			Qtd	Total	Qtd	Total	Qtd	Total	Qtd	Total
Raspberry Pi	32,00 €	un	1,0	32,00 €	14,0	448,00 €			15,0	480,00 €
SD Card	8,00 €	un	1,0	8,00 €	14,0	112,00 €			15,0	120,00 €
Power Supply	7,95 €	un	1,0	7,95 €	14,0	111,30 €			15,0	119,25 €
USB-Wireless	7,95 €	un	1,0	7,95 €	14,0	111,30 €			15,0	119,25 €
Cabo de Rede	2,95 €	un	1,0	2,95 €	14,0	41,30 €			15,0	44,25 €
Controlador 1 saídas (Digital I/o)	8,00 €	un								
Controlador 2 saídas	13,50 €	un					9,0	121,50 €	9,0	121,50 €
Controlador 4 saídas (ou 4 canais)	25,00 €	un					21,0	525,00 €	21,0	525,00 €
Jack (1 IN/1 OUT)	14,00 €	un					16,0	224,00 €	16,0	224,00 €
Interface I2c (por un)	20,00 €	un					12,0	240,00 €	12,0	240,00 €
Cablagem	0,13 €	m					150,0	19,50 €	150,0	19,50 €
RCA Converter	42,00 €	un					8,0	336,00 €	8,0	336,00 €
				58,85 €		**823,90 €**		**1.466,00 €**		**2.348,75 €**

Tabela C.4: Custo Total da Instalação do Sistema IAP.

e-commerce, de empresas de comercialização de material elétrico, de empresas de construção civil dedicadas à reconstrução e remodelação de habitações e edifícios, de empresas e gabinetes de *design* e decoração de habitações, bem como publicidade na TV, multibancos e sítios na *internet*

O produto a comercializar consiste na venda de uma licença de utilização, paga de uma só vez ou não (caso em que recebe atualizações/evoluções do produto), que permite o acesso aos serviços contratados e posteriores atualizações. O comprador apenas teria de, utilizando o número da licença adquirida, aceder ao *software* contratado através da *web* e instalar as configurações de domótica de sua habitação.

C.2 Modularidade da Instalação

Para demonstrar a modularidade do sistema IAP foram estabelecidas quatro cenários, consoante as divisões da habitação e os respetivos serviços definidos para cada divisão:

cenário I, duas divisões; cenário II, quatro divisões; cenário III, seis divisões; e, cenário IV, todas as divisões (catorze) mais o exterior da habitação.

Na tabela C.5 apresenta-se o custo total da instalação, que pode variar entre 360 e 2.350 euros, sendo o seu preço médio de 1.125 euros.

Descrição dos Componentes	Preço Unitário	Unid.	Hipótese 1 [2 divisões]		Hipótese 2 [4 divisões]		Hipótese 3 [6 divisões]		Hipótese 4 [14 divisões + Exterior]		Médio
			Qtd.	Valor	Qtd.	Valor	Qtd.	Valor	Qtd.	Valor	
CENTER				58,85 €		58,85 €		58,85 €		58,85 €	58,85 €
Raspberry Pi	32,00 €	un	1,0	32,00 €	1,0	32,00 €	1,0	32,00 €	1,0	32,00 €	32,00 €
SD Card	8,00 €	un	1,0	8,00 €	1,0	8,00 €	1,0	8,00 €	1,0	8,00 €	8,00 €
Power Supply	7,95 €	un	1,0	7,95 €	1,0	7,95 €	1,0	7,95 €	1,0	7,95 €	7,95 €
USB-Wireless	7,95 €	un	1,0	7,95 €	1,0	7,95 €	1,0	7,95 €	1,0	7,95 €	7,95 €
Cabo de Rede	2,95 €	un	1,0	2,95 €	1,0	2,95 €	1,0	2,95 €	1,0	2,95 €	2,95 €
AP's				58,85 €		176,55 €		294,25 €		823,90 €	338,39 €
Raspberry Pi	32,00 €	un	1,0	32,00 €	3,0	96,00 €	5,0	160,00 €	14,0	448,00 €	184,00 €
SD Card	8,00 €	un	1,0	8,00 €	3,0	24,00 €	5,0	40,00 €	14,0	112,00 €	46,00 €
Power Supply	7,95 €	un	1,0	7,95 €	3,0	23,85 €	5,0	39,75 €	14,0	111,30 €	45,71 €
USB-Wireless	7,95 €	un	1,0	7,95 €	3,0	23,85 €	5,0	39,75 €	14,0	111,30 €	45,71 €
Cabo de Rede	2,95 €	un	1,0	2,95 €	3,0	8,85 €	5,0	14,75 €	14,0	41,30 €	16,96 €
Dispositivos / Equipamentos				243,10 €		474,70 €		704,30 €		1.466,00 €	722,03 €
Controlador 1 saídas (Digital I/o)	8,00 €	un									
Controlador 2 saídas	13,50 €	un	1,0	13,50 €	3,0	40,50 €	3,0	40,50 €	9,0	121,50 €	54,00 €
Controlador 4 saídas (ou 4 canais)	25,00 €	un	3,0	75,00 €	5,0	125,00 €	8,0	200,00 €	21,0	525,00 €	231,25 €
Jack (1 IN/1 OUT)	14,00 €	un	2,0	28,00 €	4,0	56,00 €	6,0	84,00 €	16,0	224,00 €	98,00 €
Interface I2c (por un)	20,00 €	un	2,0	40,00 €	4,0	80,00 €	6,0	120,00 €	12,0	240,00 €	120,00 €
Cablagem	0,13 €	m	20,0	2,60 €	40,0	5,20 €	60,0	7,80 €	150,0	19,50 €	8,78 €
RCA Converter	42,00 €	un	2,0	84,00 €	4,0	168,00 €	6,0	252,00 €	8,0	336,00 €	210,00 €
Controlador 1 saída (p/Ventilação)		un									
Controlador 1 saída (p/Rega)		un									
		TOTAL:		**360,80 €**		**710,10 €**		**1.057,40 €**		**2.348,75 €**	**1.119,26 €**

Tabela C.5: Modularidade da Instalação: Quatro cenários.

C.3 Custo da Instalação do Sistema IAP

Na tabela C.6 apura-se o preço para implementação do sistema IAP, com apenas o serviço de estores em duas divisões (escritório e sala) da habitação, dado que no âmbito desta dissertação apenas foi desenvolvido este serviço.

Com base na informação recolhida verifica-se que, com apenas 360 euros, é possível instalar um serviço de controlo de estores. Na secção seguinte compara-se o custo médio de instalação do sistema IAP com os sistemas de domótica que utilizam os protocolos *standard* KNX e LON.

C.4 Custo do Sistema IAP vs Protocolos *Standard*

Nesta secção pretende-se demonstrar que o sistema desenvolvido, denominado IAP, é substancialmente mais barato que os sistemas de domótica que utilizam os protocolos *standard* (KNX e LON), sendo que o seu custo médio é de 1.125 euros.

Na tabela C.7 demonstra-se que os protocolos *standard* são cinco a seis vezes mais caros, sendo que a recolha de informação sobre instalações com protocolos KNX e LON não foi exaustiva, faltando vários componentes do sistema, pelo que o custo da instalação pode duplicar e nalguns casos triplicar.

Descrição dos Componentes	Preço Unitário	Unid.	2 divisões e um serviço	
			Qtd.	Valor
CENTER				58,85 €
Raspberry Pi	32,00 €	un	1,0	32,00 €
SD Card	8,00 €	un	1,0	8,00 €
Power Supply	7,95 €	un	1,0	7,95 €
USB-Wireless	7,95 €	un	1,0	7,95 €
Cabo de Rede	2,95 €	un	1,0	2,95 €
AP's				58,85 €
Raspberry Pi	32,00 €	un	1,0	32,00 €
SD Card	8,00 €	un	1,0	8,00 €
Power Supply	7,95 €	un	1,0	7,95 €
USB-Wireless	7,95 €	un	1,0	7,95 €
Cabo de Rede	2,95 €	un	1,0	2,95 €
Dispositivos / Equipamentos				243,10 €
Controlador 1 saídas (Digital I/o)	8,00 €	un		
Controlador 2 saídas	13,50 €	un	1,0	13,50 €
Controlador 4 saídas (ou 4 canais)	25,00 €	un	3,0	75,00 €
Jack (1 IN/1 OUT)	14,00 €	un	2,0	28,00 €
Interface I2c (por un)	20,00 €	un	2,0	40,00 €
Cablagem	0,13 €	m	20,0	2,60 €
RCA Converter	42,00 €	un	2,0	84,00 €
Controlador 1 saída (p/Ventilação)		un		
Controlador 1 saída (p/Rega)		un		
		TOTAL:		**360,80 €**

Tabela C.6: Custo *Standard* do Sistema IAP: Duas Divisões e Um Serviço.

Descrição dos Componentes	Unid. Medida	Qtd. Média	SBC (médio)		KNX		LON	
			Pr.Unit.	Total	Pr.Unit.	Total	Pr.Unit.	Total
CENTER				58,85 €				
Raspberry Pi	un	1	32,00 €	32,00 €				
SD Card	un	1	8,00 €	8,00 €				
Power Supply	un	1	7,95 €	7,95 €				
USB-Wireless	un	1	7,95 €	7,95 €				
Cabo de Rede	un	1	2,95 €	2,95 €				
AP's				338,39 €				
Raspberry Pi	un	6	32,00 €	184,00 €				
SD Card	un	6	8,00 €	46,00 €				
Power Supply	un	6	7,95 €	45,71 €				
USB-Wireless	un	6	7,95 €	45,71 €				
Cabo de Rede	un	6	2,95 €	16,96 €				
Dispositivos / Equipamentos				722,03 €		4.228,00 €		
Controlador 1 saídas (Digital I/o)	un		8,00 €		108,12 €		- €	
Controlador 2 saídas	un	4	13,50 €	54,00 €	- €		- €	
Controlador 4 saídas (ou 4 canais)	un	9	25,00 €	231,25 €	256,00 €	2.368,00 €	- €	
Jack (1 IN/1 OUT)	un	7	14,00 €	98,00 €	- €		- €	
Interface I2c (por un)	un	6	20,00 €	120,00 €	310,00 €	1.860,00 €	- €	
Cablagem	m	68	0,13 €	8,78 €	- €		- €	
RCA Converter	un	5	42,00 €	210,00 €	- €		- €	
Controlador 1 saída (p/Ventilação)	un		- €		310,00 €		- €	
Controlador 1 saída (p/Rega)	un		- €		130,00 €		- €	
	TOTAL:			**1.119,26 €**		**4.228,00 €**		

Tabela C.7: Custo Médio do Sistema IAP vs Protocolos *Standard*.

Todos os preços foram recolhidas via *internet,* sendo que uma procura mais exaustiva no mercado pode reduzir os custos apresentados nas tabelas.